ČASOPIS ZA INFORMATIKU I RAČUNARSTVO

računari

52
JUL/AVGUST 1989.
CENA 10.000

MAGMEDIA DISKETE S TRI-ŠOPOVIMA ŠIROM ZEMLJE

LOKALNE MREŽE
QUICK BASIC 4.5

MALA ST BIBLIOTEKA
GEM
ZA PROGRAMERE

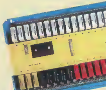


OS/2

gorenje procesna oprema

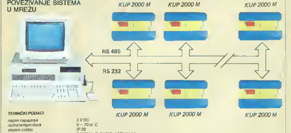
***Gorenje Procesna oprema,** p o*
Partizanska 12
63320 Titovo Velenje
Jugoslavija
Telefon (063) 853 321
Telex 33616 yu sogor

KONAČNA UPRAVLJAČKA STANICA KUP 2000 M (MINIMALNA)



Konačna upravljačka stanice KUP 2000 M nemenjene je za automatizaciju uređaja i manjih industrijskih procesa Osnovna jedinica konačne upravljačke stanice KUP 2000 M zahvata 32 digitalna ulaza ili izlaza, 12 analognih ulaza i 2 analogna izlaze
Ulazno izlazni prostor može se postavljati programski što omogućava izvenrednu prilagodljivost sistama za različite uređaje Prednost predstavlja i proizvoljni izbor ulazno-izleznih elemanata (solid state raleje) u odnosu na naponske nivoe i potrošnju na potrošačima
Osnovna jedinica sisteme KUP 2000 M može se povezati paralelno sa još tri jedneke jedinice u smislu povečanja periferijskih priključake Jedinice međusobno komuniciraju serijski neovisno jedne od druge preko RS 485 standarda

POVEZIVANJE SISTEMA U MREŽU



TEHNIČKI PODACI

napon napajanja	5 V DC
radna temperatura	0 – 70 st C
stepen zaštite	IP 20
programski jezik	PASCAL C BASIC ASEMBLER
programska memorija	64 K (RAM EPROM EEPROM)
vreme izvršenja 1 K programa	20 ms
hardverski tajmeri	/
softverski tajmeri	16
ulazi za brojače	2
softverski brojači	16
priključenje inteligentnih I/O jedinice	industrijski terminal
monoflop	16
komunikacija	RS 232 RS 485
programski alati	programski paket FEPRO
max broj dig priključaka	32 (proizvoljnih karakteristika)
max. broj anal ulaza	8 – 8 bitni
	4 – 12 bitni
max broj anal izlaza	2 – 8 bitna
max dimenzije	300 × 400 × 80 mm
max težina	3 kg

Konačne upravljačka stanice KUP 2000 M možemo povezivati u mreže preko RS 232 ili RS 485 komunikacijskog standarda

Programiranja sistema moguće je pomoću PC računara na koji se instalira željeni programski paket (FEPRO, ASEMBLER, TURBO PASCAL itd) Najpogodniji za upotrebu ja paket FEPRO koji omogućava najlakše programiranje Industrijskih procese Ulazni jezik je proširena relejske šema, a izlazni izvršni kod Po završenom programiranju program možemo preko lokalne mreže preneti u sistem ili ga zapisati u EPROM

računari 52

Časopis za informatiku i računarstvo
YUISSN 0352—7271
Izdaje BIGZ
jul/avgust 1989 — cena 10 000 D

Umetak na 16 strana

GEM za programere

str. 31—46



Naslovna strana Duško Despotović

Dovidenja u septembru!

Dragi čitaoci,

Leto je — kako nam pokazuju dosadašnja iskustva — vreme koje većina naših čitalaca koristi za odmor od škole, fakulteta, rednog meste. Odlučili smo zbog toga da broj koji je pred vama (52/jul '89) „pokrije" i avgust.

Pradah od mesec dana iskoristićemo da pripremimo nove iznenađenja, nove rubrike, nove akcije

Sledeći broj „Računara" pojaviće se u prodeji 1 septembra

Prijatan odmor

Redakcija

Izdaje i štampa
Beogradski izdavačko-grafički zavod
11000 Beograd
Bulevar vojvode Mišića 17

Generalni direktor
Dobrosav Petrović

Zamenik generalnog direktora
Antun Martić

VD direktor Novinskog sektora
Aleksandar Badanjak

Glavni i odgovorni urednik
Jovan Regasek

Zamenik glavnog i odgovornog urednika
Esad Jakupović

Marketing
Mirjana Todorović
Sergija Marčenko

Stručna redakcija
Žarko Berberski (programiranje) Vesna Ćosić (aktuelnosti) Voja Gašić (programiranje) Slobodan Perović (igra) Dejan Ristanović (programiranje i sistemski softver) Jovan Skuljan (programiranje) prof dr Dušan Slavić (matematika i numerička analiza) Nevenka Spalević (obrazovanje) Anđelko Zgorelec (dopisnik) Zoran Životić (stono izdavaštvo)

Stalni saradnici
Vladan Aleksić Žarko Berberski Viktor Cerovski Zoran Cvijetić Vesna Ćosić, Dušan Dimitrijević Voja Gašić Vladimir Janković Željko Jurić Dalibor Lanik Branko Marović Bora Milanković Slobodan Perović Dejan Predić Andrija Radović Dejan Ristanović Duško Savić Dušan Slavić Jovan Skuljan Nevenka Spalević Vlada Stojiljković Saša Svitica Žarko Vukosavljević Anđelko Zgorelec Zoran Životić

Izdavački savet
Vlado Bjelić Drago Indić Esad Jakupović Dragoljub Jakić Zoran Marković Antun Martić dr Draško Milićević dr Ljubomir Radanović Branko Rakić Jovan Regasek Dejan Ristanović dr Milvoje Simonović dr Dušan Slavić dr Dragan Uvalić, Dragoljub Vasić

Adresa redakcije
11000 Beograd
Bulevar vojvode Mišića 17/III

Telefoni
653-748 (redakcija)
650-526 (prodaja)
651-793 (propaganda)
648-140 (marketing)

Teleks
11855 BIGZ YU

Telefaks
(011) 651-641

Pretplata za zemlju
Za jednu godinu 120 000
Za šest meseci 60 000
(na žiro-račun RO BIGZ 60802-603 23264)

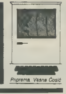
Pretplata za inostranstvo
Za jednu godinu 240 000
odnosno 19 USD 35 DEM 30 CHF 11 GBP 121 FRF
(na devizni račun RO BIGZ 60811-620-16101-620701-999--03377)

Zbog usporenog prenošenja uplate preko banke molimo pretplatnike da nam posle svake nove uplate odmah pošalju foto-kopiju uplatnice

Na osnovu mišljenja Republičkog sekretarijata za kulturu broj 413-77/72-03 i „Službenog glasnika" broj 26/27 ovo izdanje oslobođeno je poreza na promet

Rukopisi se ne vraćaju

Šta ima novo



Priprema: Vesna Ćosić

Hardver

Jedna traka za sve



„Tecmar" je objavio bekap sistem koji radi sa PS/2 običnim PC jima i „mekintošem II" Tri vrlo različite arhitekture podržane su jednim brzim sistemom trake Kapacitet je od 60 do 150 MB Razlikuju se samo adapteri za svaki od sistema posebno Adresa Tecmar Inc ,6225 Cochran Road -Solon Ohio 44139 3377

Kampjuter za bezbednost

„Immune System je DOS 286 kompjuter projektovan za bezbednost od strane jedne od najvećih kompanija koje se bave metodama bezbednog rada kompjutera Zajedno sa nekim stvarima koja se mogu očekivati od standardnog klona kao što je l MB RAM-a 1 2 Mb flopi disk jedinica l 40 MB hard disk jedinica tu je i mogućnost da se spreči unos u sistem ili izvršavanje neautorizovanih EXE i COM fajlova. Takođe je razvijen i modemski paket koji osigurava i šifrira komunikacije U isti mah obezbeđuje l evidenciju korišćenja računara evidenciju pokušaja pristupa sistemu l još 25 drugih sigurnosnih mera Osiguran je i sat tako da samo određeni korisnici mogu da postavljaju ili izmene vreme Cena 2995 USD Adresa American Computer Security Industries 122 Blue Hills Court Nashville TN 37214

Modul za akviziciju podataka

ADAC 5500MF jus jeftin modul za prikupljanje podataka smešten na polovini AT karte Ima tri programibilna tajmera koji mu omogućuju da se koristi za generisanje impulsa *timing* i aplikacije sa brojačem

Ima osam analognih ulaza multipleksiranih na 12-bitni A/D konvertor Hardversko vreme konverzije je 40 µs i može se postići propusna moć od 25 kHz. Svaki modul ima l 16 ulazno izlaznih linija na TTL nivou Pridruženi Lab Tech Acquire softver dopušta prihvat podataka na 4 od 8 analognih ulaza l jednom digitalnom ulazu

Cena 295 USD Adresa ADAC Corp.70 Tower Office Park Woburn MA 01801

LaserJet umesto plotera

Ploter u kertridžu je kompletna emulacija HP 7475A plotera i implementira HP grafički jezik (HPGL) na HP LaserJet II printeru Proizvođač tvrdi da se kompleksni crteži mogu dobiti više od desel puta brže nego na HPGL ploterima Kartica se smešta u slot za fontove printera i kompatibilna je sa većinom CAD i ostalim grafičkim softverom Između ostalog tu je i mogućnost linije različite debljine skaliranje po x i y osi generisanje luka i kruga rotacija višestruki uzorci za popunjavanje i senčenje i prilagođavanje veličini papira

Cena 395 USD Adresa Pacific data Products 6404 Nancy Ridge Dr San Diego CA 92121

SPARC za vaš PC

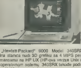
SP 1 je 20 MHz SPARC (Scalable Processor Architecture) sistem na AT karti Isporučuje se zajedno sa C i FORTRAN kompajlerima izvedenim iz onih koji se koriste na Sun-4 radnim stanicama SPARC čip je kombinovan sa „Texas Instruments" TI 8847 FPU i 8 MB DRAM-a Svaki SPARC čip ima 136 32-bitna registra i 8 prozora sa po 24 registra Proizvođač „Definicon Systems" ističe da sistem postiže 12-MIPS-a Kada se instalira koprocesor može da pristupi grafičkim karticama serijskim i paralelnim portovima A/D konvertorima i ostalim periferalima koristeći 80286 ili 80386 CPU na osnovnoj ploči kao kontrolera ulazno izlaznih operacija

Pridruženi softver prevodi DOS komande u Unix format koji očekuju Sun-ovi kompajleri Na taj način dobijate DOS koji vam je blizak i funkcionalnost rada u Sun-ovom okruženju

Cena 5495 USD sa 8 MB 7495 USD Adresa Definicon Systems Inc 1100 Business Center Cir Newbury Park CA 91320

Pre nego se eventualno odlučite na ovu skupocenu varijantu razmislite nije li bolje rešenje nabavka Sun ove radne stanice

HP sistem podržava X Windows



„Hewlett Packard" 9000 Model 340SRX radna stanica nudi 3D grafiku sa 4 MIPS performansama na HP UX (HP-ova verzija Unix a V) operativnom sistemu 340SRX takođe podržava X Windows paket

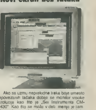
Mozak ove radne stanice se sastoji iz 68030 mikroprocesora povezanim sa 68882 FPU oboje rade na 16 7MHz Ima 4MB memorije (proširivih do 16 mega)

Pored X Windows-a komunikacije obuhvataju i RS-232C port i dve opšte upotrebljive interfejs basa (GPIB) a mogu se izabrati ili AUI LAN interfejs ili ThinLAN interfejs Podržani protokoli uključuju ARPA Berkeley TCP/IP Ethernet IEEE 802 3 I NFS

Sistem ima 16-inčni kolor monitor sa 1280 sa 1024 piksela rezolucijom Postoje 8 3D frejm bafer ravni (moguće i do 24) i 256 boja (moguće do 16 7 miliona) Sistem takođe uključuje l Starbase Graphics Library

Cena je 14500 USD a adresa Hewlett — Packard 19310 Pruneridge Ave Cupertino CA 95014 tel 800-752-0900

Novi ekran bez tačaka



Ako se uzmu neprekidne trake boje umesto nepovezanih tačaka dobije se monitor visoke rezolucije kao što je „Sei Instruments CM-1430" Kao što se može videti manje je tamnih mesta između traka nego između tačaka — tačno 33% manje Manje tamnog prostora znači da više elektrona pade na ekran a više elektrona znači dublje boje belje belo oštrije linije i čistija slova Sve to kada se kombinuje sa jednomlaznom trinitron cevi tačkom od 0 26 milimetara VGA kompatibilnošću i do 1024 X 768 rezolucijom dobija se ekran neuporediv u jasnoći preciznosti i lepoti Adresa Seiko Instruments USA Inc PC Products Division 1144 Ringwood Court San Jose CA 95131

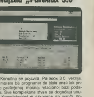
Tržište

Jeftiniji PC

Pojava 33 MHz 386 imaće svoje posledice na cene sistema sa 386 na nižim učestanostima kao l na cene sistema sa 286 Jake 386 -ice idu dole za 10—15 odsto Analize upućuju na pad cena za 8 do 25 odsto tokom ove godine u ovom srednjem delu PC tržišta Na američkom tržištu prosečna cena 286 sistema je sa 2000 USD u 1988 pala na oko 1500 u 1989 Delom je to i posledica nešto povoljnije situacije sa memorijama (trend pada cena po bitu posebno kod 1Mbit-nih čipova)

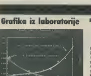
Softver

Najzad „Paradox 3.0"



Konačno se pojavila Paradox 3 0 verzija. Ne morate biti programer da biste imali lak pristup prefinjenoj moćnoj relacionoj bazi podataka. Sve kompleksne stvari se događaju unutra Kompleksnost je sakrivena iza jasnih prirodnih čak intuitivnih puteva koji vam olakšavaju život Sa „Paradoxom" nema nejasnoće i nepoznatog Ne treba vam armija ljudi za tehničku pomoć i — što je najvažnije — ne morate biti programer da biste bili profesionalac

Grafika iz laboratorije



Poslednja verzija 3.0 SideWnle Plus-a je namenjena naučnicima i inženjerima koji nameravaju da kreiraju grafike i karte za objavljivanje. Nova verzija nosi preko sto novih mogućnosti, uključujući indeksi hardverskih fontova, poodljšanu podršku za odje više tipova i debljina linija, kao i više uzoraka za popunjavanja.

Verzija 3.0 sada podržava 16 *foreground* i 16 *backgrond* odja. Uzorci za popunjavanje uključuju više ploiarskih uzoraka i polutonova za HP LaserJet i druge laser printere. Mogućnost crtanja jednačina daje crtanje bilo koje jednačine u obliku y=f(x). Mogu se definisati do 12 jednačina i kontrolisati broj generisanih tačaka za dobijanje glatke krive. Curve-fiting omogućava crtanje glatkih krivih kroz sve vaše tačke podataka, mogu se specificirati debljina linije i uzorak za svaku krivu. Takođe mogu se učitati ASCII podaci u graf, te uneti TIFF, PCX, CGM fajlovi i konstili za specijalne efekte na kartama. Program može da odredi srednju vrednost, standardnu devijaciju i standardnu grešku u toku učitavanja ASCII fajla.

Cena 445 USD. Adresa: Advanced Graphics Software, 333 West Maude Ave., Suite 105, Sunnyvale, CA 94086

Procesna kontrola

ULTRABRZO UZORKOVANJE NA OSCILOSKOPU

Inženjerima i naučnicima koji zahtijevaju ultra brzi i jednostruki signal firma „Hewlett-Packard" pruža jedno korisno poodljšanje. Područja primjene su laseri, visokoenergetska fizika, poluvodiči velike brzine i digitalni dizajn, a reč je o novoj test garnituri HP 54114A za digitalni osciloskop HP 54111D koja omogućuje najveću brzinu uzorkovanja među osciloskopima. Za prilagođavanje nove test garniture na osciloskop je dodat novi firmver. Građevinje također pridodaje neizbrisivu memoriju velnog oblika.

Dodatna brzina uzorkovanja povećava maksimalnu jednostruku širinu frekventnog pojasa HP 54111D od 250 MHz na 500 MHz i osigurava hvatanje glitcheva sve do 500 ps. Filteri širine širine pojasa osiguravaju vertikalnu rezoluciju od 6—7 i 8 bita. HP 54111D ima 8K memoriju velikog oblika po kanalu. U jednom kanalu sa 2 giga uzorka u sekundi konfiguracije memorija se udvostručava i dopušta sve do 8 µs informaciju koja se zadržava za kasniju analizu okolnosti koje su prethodile izbacivanju podataka.

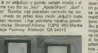
Novi aparat se koristi sa bilo kojim HP 54111D digitalnim osciloskopom. Instrumenti proizvedeni prije povećanja iziskuju ugradnju garniture koja se sastoji od PC ploče koja sadrži ugrađeni firmver i novu neizbrisivu memoriju. Aparati proizvedeni nakon uvođenja test garniture obuhvaćaju ugradnju samo HP 54114A da se postigne mogućnost od 2 giga uzorka u sekundi.

Andrija Šijak

Komunikacije

Nema rata među mašinama

Izgleda da svaki put kada se zajedno nađu PC i „mekintoš" — počinje bitka. Ako jedan kaže dan, drugi kaže noć. I odmulo. Međutim, nešto se promenilo. Dobra vest je da su ljudi sa TOPS-om sredili ovu svađu jednom zauvek. A bilo je lako jednostavno. Rezultat je da sada samo instalirate TOPS softver u „mekintoša" i PC, ubacite u njih mrežnu karticu, dodate malo telefonske žice i oni se odjednom ponašaju kao familija — i to složna. Posle toga, naime, oni razmenjuju fajlove, dele periferije i prenose podatke iz „Lotusa" u „Excel", „dBase III", „dBase Mac", većinu ostalih PC i Mac aplikacija. Ili se uključuju u gomilu ostalih mreža i sistema kao što su „Vax", „AppleShare", „Sun" i „Vax". Štaviše, nije potreban centralni fajl server, može se jedan lako može uključiti kada posao nalesie. I nije potrebna nikakva posebna obuka. Stvarno! Adresa: TOPS, 950 Marina Village Parkway, Alameda, CA 94510



Za integracija Fax-a



„Facsimile Server FaxPress" firme „Castelle" uključuje dva Motorolina 68000 mikroprocesora, jedan koji radi kao fax procesor, a drugi kao procesor mreže. Dobijate i 5 MB RAM-a, serijski port, paralelni port i Ethernet ploču.

Cena 3650. Adresa: Castelle, 2540 Mision College Blvd., Suite 102, Santa Clara, CA 95054

Tržište

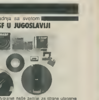
Prva računarska trgovina

U Ljubljani (Igriška 14) otvorena je prva specijalizovana trgovina za mikroračunarsku i birotehničku opremu. Aktivnost trgovina obuhvata prodaju različitih mikroračunarskih sistema, dodatne mikroračunarske opreme (perifernih jedinica), različitih mikroračunarskih dodataka, pojedinih mikroračunarskih sklopova, vlastite aplikativne programske opreme, tuđih programskih paketa, elektronske birotehničke opreme, kompletnu servisnu delatnost za mikroračunarsku i birotehničku opremu, školovanje i savetovanje prilikom nabavke opreme i programa i inženjering za projektovanje i poremanje preduzeća mikroračunarskom i birotehničkom opremom.

Trgovina predstavlja rezultat saradnje sa stranim partnerom, firmom „Retzl" iz Graca, obrtnim firmama „Orbit" i „Hibit" iz Celja i „Finomehaniku" Celje, sa kojima ona formira i mešovito preduzeće.

Saradnja sa svetom

BASF U JUGOSLAVIJI



Otvaranje naše zemlje za strana ulaganja, stvaranje pravnih okvira za formiranje mešovitih preduzeća, izjednačavanje stranih partnera sa domaćim, garantovanje transfera valuta u matičnu zemlju i druge već usvojene ili najavljene promene u ekonomskoj sferi privlače veliku pažnju u svetu. Poznata zapadnonemačka korporacija BASF je svoje zanimanje izrazila na reprezentativan način, posetom našoj zemlji dr Inga Peckea (Paetzke), člana poslovodnog odbora, i dr Klausa Kanclera (Cantzler), direktora zaduženog za Jugoslaviju i Istočnu Evropu, izvanrednim prijemom za veliki broj zvanica u „Interkontinentalu" i izuzetno dobro posećenom konferencijom za štampu u beogradskom predstavništvu BASF-a (koje je u uzornom stilu organizovala Agencija „Duga" — BIGZ) te sličnim pandanima održanim u Zagrebu. O značaju posete svakako nešto govori i činjenica da je goste primio i potpredsednik SIV-a Aleksandar Mitrović.

Kao formalni povod za ovaj zanimljivi i sjajni nastup ugledne firme poslužili su obeležavanje 40 godina saradnje sa Jugoslavijom i završetak renoviranja i modernizovanja predstavništava u Beogradu i Zagrebu, ali nije se krilo da su rukovodioci BASF-a došli i da iz prve ruke dobiju informacije o liberalizaciji naših propisa i mogućnostima za zajedničke poduhvate.

Osnovan 1865. godine, BASF je danas jedno od 30 najvećih hemijskih preduzeća na svetu sa ostvarenom godišnjom prodajom od bizu 50 milijardi nemačkih maraka. Na promet sa Jugoslavijom otpada skroman, iako obema stranama važan deo od 200 miliona (prema milion maraka iz 1949. godine) — koji bi u dogledno vreme mogao višestruko da se uveća ako bi došlo do zajedničkih ulaganja ili formiranja mešovitog preduzeća. Bez obzira što ne postoji nikakav gotov koncept, što ostaju otvoreni razni problemi, od kojih je najteži naša nelikvidnost, za optimizam sada sigurno ima dosta osnova — što ova značajna poseta i sama po sebi potvrđuje.

Firma BASF se bavi proizvodnjom različitih grupa proizvoda s područja osnovnih materija, disperzije i oplemenjivanja u raznim granama industrije, zaštite bilja, tehničkih plastičnih materija, specijalnih hemikalija — sve ukupno preko 6 000 proizvoda za hemijsku industriju, najpre preradivačku, automobilsku, grafičku, tekstilnu, kožnu, građevinsku i elektro industriju i poljoprivredu. Za ljude koji se bave računarima naročito je zanimljivo da je BASF proizveo prve magnetne trake u svetu. Nije ni čudo da u toj oblasti kao i u proizvodnji raznih vrsta diskela gotovo da nema premca. Oblast informatike postaje za ovu firmu sve važnija, što najbolje potvrđuje sve širi asortiman proizvoda kompjuterskih medija za specijalna primena, štampača, sistema za memorisanje i (sa firmom „Siemens") samih računarskih sistema pod imenom „Comparex".

Sve u svemu, partner sa kojim se može lepo i korisno sarađivati.

Esad Jakupović

Božidar Travica

Daj 100 dolara!

Bauk parničenja kruži američkom softvarskom industrijom: „Epl", „Ešton-Tejt" i „Lotus" namerili su da preko suda isteraju svoja autorska prava, a zaštita intelektualne svojine predstavlja glavnu temu razgovora u programerskim krugovima.

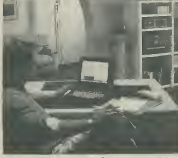
Prva polovina ove godine protiče u rasplamsavanju argumenta i strasti u vezi sa zaštitom intelektuelne svojine. I nisu samo softverski džinovi na sceni. Naprotiv. Vetru kao da više raspiruju mali proizvođači softvera, koji u sve većem broju, počinju da mašu potvrdama o patentiranom softveru. Naravno, mahanje je samo propratni čin onog glavnog — zahteva da se deli profit s onima na koje sumnje pada da su neovlašćeno posegnuli za zaštićenim rešenjima. I stvari se toliko zahuktavaju da se čini da se softverska scena sa tržišta i izložbi seli — u Justicijina dvore.

Prošlog meseca patentno pravo dobila je mala kalifornijska programerska firma „Kvoterdek ofis sistems". Reč je o softveru za rad s različitim programima s kojima se istovremeno komunicira u odvojenim prozorima.

Drugi slučaj je takođe iz „Zlatne zemlje". Izvesni Pol Hekel uspeo je da patentira aplikaciju napravljenu korišćenjem poznatog „Eplovog" programerskog alata „hajperkard" (Hypercard). Zatim je počeo da razašilje ponude i malim i velikim korisnicima „Eplovih" računara, nudeći svoju robu za 100 dolara po kopiji. Novinarima je izjavio da svako ko koristi „hajperkard" ugrožava njegov patent.

Potpredsednik firme „Vordperfekt" (Wordperfect) Pit Peterson (Pete Peterson) kaže da mu svakog meseca pristigne najmanje jedan zahtev od vlasnika patenata koji uveravaju da je firma iskoristila njihove patenta i traže pare.

Saznaje se da u američkoj ustanovi za registrovanje pateneta postoji gomilica zahteva malih proizvođača softvera. Rok za dobijanje patente je oko tri godine. Zahtev za prvopomenuti patent datiran je s 1984. godinom. I kada se svi ovi podaci saberu, izlazi da ima mesta za zvona za uzbunu ili — za veselje, zavisno od toga s koje se strane stvar posmatra.



Suprotni tabor, međutim, poteže drugačije argumente. „Zamislite šta bi se dogodilo kada bi neki kompozitor patentirao niz note u sledu i zato tražio odštetu od svakog koji pravi muziku što podseća na njegovu temu", veli pomenuti Peterson. Možda bi, ako je u životu, i potomci sobarice koje je zakucala na vrata Betovenove hotelske sobe, davši maestru inspiraciju za osnovu temu V simfonije, mogli sada da traže odštetu od bezbrojnih plagijatora kroz vekove — pada mi nešto na pamet.

No, šalu na stranu. Tema je ozbiljna i izgleda da se teško nešto ubedljivo može zaključiti samo na osnovu poređenja s ostalim oblicima intelektuelne svojine. Kritičari patentske trke kažu da su patenti često bili izdavani za delove programa koji su odavno postali standard u softverskoj industriji. Šta, na primer, ako se pod okriljem zakonske zaštite nađe kôd za pokretanje kursora po ekranu, ko će sve i koliko morati da plaća vlasniku patenta? — pitaju neki.

Istini za volju, za sada se iz izjava kritičara pre može pročitati strah od sudskog rešavanja softverskih odnosa, nego li odbojnost prema principu patentne zaštite. Očigledno je nepoverenje u kompetentnost sudskih organa, ali i u sve ono što se zbiva kada se sud osloni na mišljenje eksperta. Dakako uz neizbežne advokate.

Valja naime imati u vidu širu američku perspektivu, tj. glas koji prati advokatski ceh. On ovde, najblaže rečeno, nije omiljen. Pučanstvo na advokate gleda kao na mutikaše koji od ništa prave nešto i pomoću marifetluka zgrću ogromnu lovu. Slično kao u Engleskoj u doba „Guliverovih putovanja". A advokati su se već oglasili. Kažu, skoro svaki program ili delovi programa mogu biti predmet patentiranja!

Hoće li bauk parničenja postati svakodnevica — ostaje da se vidi. Zdrava logika, međutim, govori da će u slučaju otvaranja novog, atraktivnog tržišta za advokate, tanji kraj izvući oni koji sada usijavaju problem — mali, čija je platežna moć takođe mala.

Filipič u Americi

Deo izveštaja posvećenog kratkim vestima započinjem s novošću koja godi jugoslovenskog posvećenika u računara. U ovdešnjim bibliotekama, u knjižarama

Advokati trljaju ruke

Razumljiv je zahtev da se zaštiti intelektualni produkt. Ako na to pravo imaju pisci, kompozitori — zašto ne bi imali i programeri? Uz to, zašto da samo softverski džinovi uživaju Justicijinu naklonost, a ne i samostalni programeri ili male firme? IBM, na primer, godišnje za patentiranje prijavi oko 200 sopstvenih softverskih rešenja, što je trećina od njegovih godišnjih patentnih prijava.

dakle i od pre sada se može naći knjiga našeg kompjuterskog stručnjaka Bogdana Filipiča iz Instituta „Jozef Stefan". Knjiga o programiranju u PROLOG-u zove se „Prolog User's Handbook. A Library of Utility Programs", a izdala su je firme „Ellis Harwood" i „Halsted Press", 1988. U listi ostalih naslova izdavača, pored Filipičevog, upisano je ime Nade Lavrač kao koautora. Knjiga je dobro opremljena i na pravi način predstavlja deo iskustva do kojih su došli Filipič i kolege iz „Stefanove" laboratorije za veštačku inteligenciju.

Uzgred, sasvim nevezano za temu ovog hvale vrednog spisateljskog poduhvata, pada mi na pamet takođe dobra knjiga Ričarda Forsajta (Richard Forsyth) i Krisa Najlora (Chris Naylor) „Autostoperski vodič za veštačku inteligenciju" (The Hitch-Hiker's Guide to Artificial Intelligence). Trebalo bi da je bude i u Jugi, budući da je izdata pre dve godine.

Sa opuštenošću čoveka koji je napustio predavački posao na londonskoj politehnici i sada vodi svoj biznis projektujući ekspert sisteme, Forsajt piše razumljivo i privlačno. Što je još važnije, razbija zastrašenost masa pred tajnama veštačke inteligencije, pokazujući primerima da je reč o običnom programiranju sa nešto drugačijih pozicija.

Upotreba LISP-a ne mora značiti i da programer tera mašinu da „misli", objašnjava Forsajt. Njegova knjiga obiluje primerima kako se obična Basica može koristiti za rešavanje klasičnih problema iz oblasti veštačke inteligencije. Njegovo mišljenje o PROLOG-u može da naljuti specijaliste. Ako se veli Forsajt, Japan opredelio da pretežni ulog stavi na sisteme koji slede filozofiju PROLOG-a, Evropa i SAD mogu neko vreme da odahnu: budućnost nije u tom pristupu.

Uskoro će se na tržištu SAD naći klonovi „epla". Reč je o Tajvancu, tj. produktu firme „Akkord Technology" koja je napravila klon „meka plus". U SAD-u će ga prodavati pod nazivom „Džonatan" za oko 300 dolara jeftinije od originala. „Eplovi" obožavaoci u ovome vide znak ostvarenja svoga sna koji je inspirisan istorijom IBM-ovog PC-a: što više mašina na tržištu, to niža cena, to dostupnije mašine („Eplove") kompjuterskim masama.

Ubrzo pošto je početkom aprila „Intel" najavio 80486 čip, bosovi firme UNIX su izjavili da će njihovi korisnici prvi imati na raspolaganju moći novog procesora. On će moći da opslužuje 2—3 puta više korisnika nego 32-bitni čip (do 64) i izvršavaće 15 miliona instrukcija u sekundi (32 bitni može „samo" 8).

Dobra vest za fortranovce: „Majkrosoft" je počeo da isporučuje novu verziju popularnog PC kompajlera s oznakom „5.0". Novost je poboljšani dibager, linker za DOS aplikacije (ne samo za OS/2 kao što ima FORTRAN 4.1) i grafička rutina.

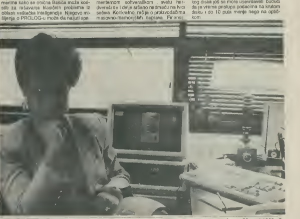
Traganje za memorijom

I da završna redova posvetim hardverašima. Drugačije od trenda u komplementarnom softveraškom svetu, hardveraši se i dalje srčano nadmeću na ivici sečiva. Konkretno, reč je o proizvođačima masovno-memorijskih naprava. Finansijski, kolo vode proizvođači krutih diskova mada ni predstavnici klasične trake ne zaostaju. Obe tehnologije međutim sve više trpe konkurenciju od strane naprednijih medija.

Trakama konkuriše uljez iz audio sektora, tzv. DAT (digital-audio tape). U odnosu na četvrt-inčne trake DAT znači veće spremište na manjem prostoru. Kao i svaka nova tehnologija, DAT je još skup. No, po megabajtu memorije već je jeftiniji od klasične trake.

Krutim diskovima pak konkurišu optički diskovi. Glavnu reč vode tri tehnologije: CD ROM (compact disk read-only memory), WORM (write once/read many) diskovi i MO (magneto-optic) diskovi. Svaki od ovih tipova diskova ima svoje prednosti, no nedostataka najmanje imaju MO-diskovi. Prva dva tipa diskova obezbeđuju ogroman memorijski prostor, ali imaju i bitno ograničenje — nemogućnost menjanja podataka. To je dobro za neka poslove (bibliotečka i pravna, npr.) ali ne i za privlačenje masovnog kupca. MO diskovi imaju 200—500 megabajta memorijskog prostora i podaci se na njih mogu pisati i brisati kao i na kruta diskova.

Ukrštanjem performansi WORM i MO diskova može se dobiti memorijska naprava budućnosti. No, proizvođači krutih diskova imaju vremena da amortizuju proizvodni park i blagovremeno se uključe u novi trend. Naime, tehnologija optičkog diska još se mora usavršavati budući da je vreme pristupa podacima na krutom disku i do 10 puta manje nego na optičkom.



OLIVETTI M386

Brzi Gonzales

Sa nastrpljenjem smo očekivali da probamo ovu mašinu, jer nam je to prva mašina koja koristi 32-bitnu magistralu i koja radi na 20 MHz koju smo imali prilike da prikažemo.

Dejan V. Veselinović

„Olivetti 386" predstavlja vrh proizvodne game renomirane italijanske firme „Olivetti" sa izuzetkom kapaciteta tvrdog diska koji po želji može biti još veći od onoga koji smo mi probali i mogućnosti da ceo aranžman dobije zapakovan u „tower" kutiju. Srce mašine čini INTEL iAPX 80386-20 procesor sa postojećem za koprocesor i memorijom montiranom na standardan format AT karte za proširenja. U računaru se takođe standardno nalaze i po jedan serijski i paralelni vaznik. Spolja kutija mašine deluje veoma lepo (mada je šablonski četvrtasta) i ne preveliko manja veličina je postignuta ograničavanjem prostora na tri poluvisinska nivoa za tri spoljna pristupačne jedinice spoljne memorije. Boja mašine i tastature je smirujuća braon, a na tastaturi je izveden efekat memera iznad gornjeg reda tastera. Obrada je zaista izvanredna i daleko odskače od proseka.

Priča za sebe

Tastatura je sasvim druga priča. Standardnog je oblika i rasporeda proširene AT tastature i u celini deluje sasvim pristojno ali ispod nivoa celog računara. Ono što je čini zaista jedinstvenom bar po našem iskustvu jeste osećaj prilikom pritiska na tastere. Do sada smo probali mnogo tastatura ali ovakvu nikada ni smo videli. Na početku pritiska veoma je mekana a što više pritiskate postaje sve tvrđa. Očekivali biste da do kraja hoda postane dosta tvrda sve dok ne dođete do dna, koje bi po logici trebalo da bude čvrsto kao beton. Umesto toga, na kraju hoda kao da naiđete na neku ubi bože jeftinu gumu koja je prida i vlažna ceo osećaj je nekako razočaravajući. Istini za volju moramo priznati da sve lepo funkcioniše ali ipak morate dobro da pritisnete da biste dali kontakt. Olivettijeve tastature su slično COMPAQ-ovim uvek bile priča za sebe i uvek ste ih ili voleli ili mrzeli. Ovoga puta, međutim dileme nema ovu ćete mrzeti.

Monitor po svojim dimenzijama deluje zaista veoma ubedljivo i solidno mada ne i glomazno. Italijanski dizajn. Slika koju daje je već nešto drugo. U crno-belom modu (koji je inače i DEFAULT ili početni mod) slova su ispod kvaliteta i ole boljeg tajvanskog ili korejanskog monitora, koji sigurno i košta manje. Naime oko bele osnove slova jasno se i sa metar razdaljine vide razlivene boje posebno crvena, hipnagogični NEW PRIMITIVES umetnici će se raspakati od sreće a svi ostali od nesreće. Valjda kao kompenzacija za ovo rad sa bojama je zaista zadovoljstvo. Niti se prelivaju niti su mutne niti nejasne sem već pomenutih problema sa belom bojom. Povrh svega lačno su raspoređena po šemi IBM-a što osetno olakšava podešavanje u svim programima. Kao plus monitoru treba reći i to da ima veoma konzistentnu sliku što znači da nema razlivanja boje po uglovima niti dolazi do gubitka oštrine. Komande za kontrast i svetlost su inteligentno skrivene u prednjem donjem uglu monitora i apsolutno su pristupačne i svrsishodne odnosno imaju dobar opseg efekta.

Grafika i tvrdi disk

Grafika računara bi trebalo da bude saglasna sa VGA standardom kažemo samo „trebalo bi" više kao neku posleduju rezervu jer nismo imali prilike da testiramo računar sa obiljem programa. U testovima kompatibilnosti koje je razvio „PC Magazine" video karta je bez i jedne jedina greška prošla kroz sve testove CGA, EGA i VGA standarda uključujući i test hardverske kompatibilnosti sa VGA. Ukratko sve indicije govore da je stvarno kompatibilna. Ono što treba pohvaliti jeste da je rad karte veoma „meknu" bez ikakvih trzanja ili treperenja ili prelaznih razlivanja boja uprkos činjenici da se rad odvija veoma brzo. Zaista pun pogodak za Olivetti (posebno nakon njihovog standarda koji je bio samo njihov) jer se vratio uobičajenim vodama ali je to uradio na velika vrata.

Najzad, tvrdi disk u probnoj mašini ima nominalno 81 MB a bio je podeljen na DOS i ne-DOS deo. Pošto se radilo o poslovnoj mašini nismo ni hteli ni smeli da menjamo parametre ove podele pa smo morali da se zadovoljimo sa svega 5 MB ostavljenih DOS-u. Iz iskustva znamo da to neće bitnije uticati na brzinu rada jer smo efekat manjeg prostora kompenzovali popunjavanjem što je protivno pravilima igre. Najinteresantnije kod diska je njegova brzina ali i gotovo potpuna nečujnost. Da biste znali da se okreće morate punim uvo sasvim blizu ili dodirnuti mašinu. Na bunkte za izduvni ventilator sa zadnje strane računara shodno tradiciji Olivetti je i ovaj put ugradio ventilator velikog prečnika ali zato manjeg broja obrtaja, pa cela stvar radi bez gubitaka u efektu ali sa manje buke. Kao i ranije ventilator se nalazi potpuno van kutije odnosno montiran je na poleđini mašine.

Metodologija testiranja

Reč-dve o metodologiji testiranja i testovima koje smo upotrebili. Kao osnov poslužili su nam po našem mišljenju izvanredni testovi mašinske opreme koje je razvio američki časopis „PC Magazine". Upotrebili smo verziju 5.0 iz aprila ove godine pa možemo tvrditi da su sveži da svežiji ne mogu biti. Za testove rada sa programima upotrebili smo dva kod nas dosta dobro poznata programa, „WordPerfect" 5.0 i HARVARD GRAPHICS, verziju 2.12 opet veoma svežu. U svim navedenim testovima CONFIG.SYS datoteka je sadržala red „buffers 20" sem u slučaju testova sa tvrdim diskom jer autori testova preporučuju da vrednost međumemorije ne bude veća od 3 pa smo ih poslušali.

Rezultati svih merenja su prikazani u tabeli rezultata. Pregledom ćete primetiti što-šta. Što se merenja procesora i memorije tiče vidi se da su uverevno osetno bolja od prvog sledećeg konkurenta ZEOS 286/12 modela koji radi na 12 MHz bez stanja čekanja. Kada bi i Olivetti zaista radio bez stanja čekanja trebalo bi da bude dva puta brži od kloue prikazanog u koloni dva. takođe je jasno da je u stvarnosti M386 prosečno 49.7% brži a najmanja razlika od jedva 26.2% je upravo u pristupu DOS memoriji. Pošto AT klon zaista nema stanja čekanja jasno je da ih Olivetti i te kako ima i to po svemu sudeći podosta.

Memoriju iznad 640 kB (instaliran je 1 MB) jednostavno nismo mogli da pronađemo. Verovatno je ušlo učinjeno prilikom konfigurisanja sistema za rad u drugom operacijskom sistemu tek te memorije nigde. Inače drugi upravljački sistem koji je zauzimao preostalih 76 megabajta prostora na tvrdom disku je bio QNX. pošto nemamo nikakvog iskustva sa ovim sistemom ostaje nam da zaključimo da je on na neki način odstranio razliku između instalirane i raspoložive memorije.

Za razliku od nje tvrdi disk je bio i te kako prisutan a rezultati govore da specifikacija od 16 mS zaista vredi jer je svi rezultati potvrđuju. Dobijena vremena su zaista najbolja koja smo do sada ikada izmerili a potsecaju nas na naš AT klon sa keš memorijom. Imajući u vidu kalibar mašine i verovatnu namenu kao „file server" odnosno centralna jedinica u mreži terminala postaje razumljivo zašto treba imati tako brz disk.

Rezultati video merenja su takođe vredni i pažnje i poštovanja. Naime VGA karte su na zlu glasu zbog njihove sporosti rada (najčešće



rade u 8-bitnom režimu) mana koja se Olivetiju svakako ne može pripisati Mada su vremena ispisivanja jednog ekrana ista kao i kod Zeosa ekran sa skrolovanjem je duplo brži a neposredan pristup ekranu je čak 2 3 puta brži Brzina ovde svakako nije problem a uz kompatibilnost i dobre boje sem COMPAQ video karti mi ne znamo za bolje ilustracije radi IBM model 70—121 koji radi na 25 MHz i ima 64 kB SRAM keš memorije ima oko 2 3 puta sporije video performanse

U pravoj akciji

Rezultati merenja sa „WordPerfect" programom su mahom onakvi kakvim smo i očekivali da će biti sa jednim izuzetkom Rezultati video testa u okviru ovog programa su u najmanju ruku izuzetni Test pokazuje da je Oliveti preko pet puta brži od prvog sledećeg računara (ZEOS) i ravno deset puta brži od starog IBM PC računara Nema sumnje da je ovo rezultat pre svega veoma brze video karte a ovo je opet indikacija činjenica da su video karte često zapravo glavno usko grlo u ovoj vrsti rada (grafika) daleko više no sama brzina procesora

HARVARD GRAPHICS 2 1 je takođe sasvim lepo radio Dobijeni rezultati su veoma interesantni Prvi test („Učitavanje") zapravo ni smo mogli da obavimo iz jednog banalnog razloga HERCULES verzija uvodnog ekrana jednostavno nema pa su prethodna merenja obavljana bez nje No EGA/VGA verzija postoji a ona je osetno zadržala računar stvarno vreme je bilo 7 23 sekunde što je očigledna besmislica u ovom slučaju No interesantno je primetiti kako se čisto hardverska prednost Olivetija istopila u ovom testu i sa prednosti od 2 3 1 pala na jedva 1 21 1 Mada je tačno da kvalitet programskih video veznika utiče na ova merenja takođe je tačno da ćete vi raditi sa upravo takvim programima a ne sa testovima brzine Odnos se malo popravlja (na 1 3 1) kod složenijih slika ali je i tu ispod onoga u hardverskom testu Dobuše treba napomenuti da je umesto crno-belo Olivetl crtao mape gradova u više boja tako da je krajnji rezultat izgledao neuporedivo impresivnije no na našoj HERCULES PLUS karti

Što se klasičnih merenja tiče CORE test nas je obavestio da je brzina prenosa kontrolera i tvrdog diska 795 l kB/s LANDMARK Speed test kaže da mašina radi na 20 6 MHz, a CHIPS & TECHNOLOGIES MIPS test kaže da je OLIVETTI M380 6 93 puta brži od PC-a 2,4 puta brži od AT-a i 1 18 puta brži od COMPAQ 386 (na 16 MHz) dok apsolutan broj izvršenih naredbi u sekundi iznosi 2 39 miliona

Najzad dokumentacija za računar je na nivou onoga na šta smo navikli od Olivetija dobra i celovita Iznad svega veoma je jasna i relativno jednostavno napisana tako da je zaista svako pismen može lepo razumeti i ovo je deo tradicije kod Olivetija

Najzad kako oceniti ovaj računar? Pozitivne strane su mu odličan tvrdi disk i odličan video deo Procesorska merenja, mada dobra ipak zaostaju za klasom kojoj pripada Jedva su oko 11% bolja od COMPAQ DESKPRO 386 koji radi na 16 MHz Osnovni razlog je svakako nedostatak bilo kakve statičke keš memorije što na 20 MHz postaje nužnost a efektivno ubrzava računar za 20—30 procenata u zavisnosti od veličine i organizacije keš memorije Da stvar bude tragičnija prema nekim našim preliminarnim merenjima nove NEAT AT ploče koje rade na 16 MHz imaju performanse jedva za oko 11 % ispod nivoa M386 ali zato koštaju kudikamo manje

Iznenađenje iz kutije

Po svemu sudeći proizvođač se opredelio za najjednostavnije rešenje u vezi odnosa brzina procesora/memorije a to je da uvede jedno stanje čekanja Time se ciklus pristupa RAM u produžava na 100 nS a takvih čipova ima koliko hoćete i gde hoćete Šteta je što se ni su opredelili za preplitanje (INTERLEAVING) ako ništa više dobit bi u brzini bio šta između 9% i 15% ovako celom proizvodu nekako nedostaje sjaj, deluje pomalo bezbojno

Pa ipak ovaj računar ima jednu izuzetno važnu osobinu koja se ne vidi na prvi pogled Poznato je da OS/2 podržava 16 MB memorije što je ustupak svima nama koji imamo 80286 mašine Takođe je poznato da 80386 procesor može da adresira čitave terabajte memorije, mada mu je za sada osnovna prodajna vrlina kombinacija brzine i sposobnosti da glumi nekoliko 8088/8086 procesora u nekoliko prozora bez opasnosti da će slom u jednom prozoru izazvati slom celog sistema (kao kod 80286) Brzina procesora plus brzina tvrdog diska je jednaka nameri proizvođača da tu mašinu prodaje kao centralni računar u mreži Sve ovo imaju i svi ostali kudikamo jeftiniji proizvođači Ono što koliko je nama poznato ima samo ovaj računar (i njegova verzija za S A D AT&T 6386 WSG) jeste sposobnost da prihvate i iskoriste ne 16 već punih 48 MB RAM a i to u punoj brzini Mada je tačno da se od centralnog računara pre svega očekuje veoma brz disk ovoliko memorije ga čini samo još bržim i pogodnijim za opsluživanje većeg broja terminala

Ako je ovo bila namera kupca Udružene Beogradske Banke onda teško da su mogli kupiti bolje (kada se sve uzme u obzir) za pojedinačne korisnike smatramo ovu mašinu neprikladnom ne zato što je loša već zato što za manje pare možete dobiti daleko bolje mašine na drugim stranama Istina nećete imati servis kao Olivetijev a verovatno ni taj opšti kvalitet i podršku proizvođača, ali ćete i platiti oko 40—60% uobičajene cene

	IBM PC 4 8Mhz	AT klon 10MHz/O	ZEOS 12MHz/O	OLIVETTI 20MHz/O
Mašinski testovi				
Procesor i memorija				
1 Brzina procesora	14 06	2 60	2 23	1 82
2 Naredbe 8088/8086	32 57	5 66	5 00	3,85
80286	—	5 55	4 84	3 74
80386	—	—	—	3 68
3 Numerika bez 87	151,38	22,24	18 06	13,32
sa 87	—	—	4 84	—
4 Memorija DOS	5 98	0 77	0 71	0 61
AT (preko 1 MB)	—	12 45	10 16	(1)
LIM (3 2, 4 0)	—	—	—	—
Tvrdi disk				
1 Pristup DOS datotekama				
— male	136 48	76,17	64 98	56 81
— velike	36 98	8 97	7 16	5 52
2 DOS pristup tvrdom disku	90 92	35 52	40 72	17 58
3 Disk BIOS pristup				
— redno	36 21	15 71	8 73	4 30
— slučajno	66 21	29 01	27 01	17 82
Video	HERC K	HERC +	HERC K	OLIVETTI
1 Ekran bez pomeranja	29 86	1 21	1 01	1 05
2 Ekran sa pomeranjem	36 96	3,24	3 80	1 92
3 Neposredan pristup ekranu	16 98	4 89	4 88	2 14
Programski testovi				
Obrada teksta (WordPerfect 5 0).				
Učitavanje	63 48	11 02	14 78	10 12
Brojanje reči	372,42	73 00	66 01	49 65
Brisanje	113 94	16 06	16 19	13 99
Traženje i zamena	360 47	62 40	55 57	42 31
Izgled	54 71	37 21	28 24	5 48
Pisanje na disk	146 64	45 35	43,23	35,38
Grafika (Harvard Graphics 2.1)				
Učitavanje	8 02	3 54	3 00	< 2
Mape gredova	135 76	21 06	17 79	13 62
Poslovni grafikon (histogram)	20 34	4.65	3 80	3 14

(1) iz nepoznatih razloga, računar nije prepoznao inače postojeću memoriju iznad 640 kB.

Kompjuter za Svetog Savu

Ako bi se, daleko bilo, kojim slučajem kupole Hrama Svetog Save, sa svih četiri hiljade tona u nju ugrađenog betona, strmoglavile na putu do četrdeset metara visokog mesta stalnog boravka — njen pad bi, sasvim sigurno, dobro prodrmao Beograd. Da se to ne dogodi brine, između ostalog, i sistem za akviziciju podataka jedne privatne firme.

Radislava-Duda Vujasinović

Istorija je zabeležila da je graditeljima istanbulske Aja Sofije tek iz trećeg puta pošlo za rukom da smeste kupolu tamo gde joj je mesto. Od tada su prošli mnogi vekovi, vremena su se promenila i danas inženjeri KMG „Trudbenika" tvrde da spori podizanje kupole svetosavskog Hrama polažu iz cuga. Kupolu podiže nabu pod oblaka elektronsko hidraulični sistem Irstaničke „Prve petoletke". To je ono što se zna. U galeriji kupole smeštena su i merne trake i deformetri Instituta „Kirilo Savić" koji registruju naponska stanja. Ali tek uz komandni pult sistema za dizanje na skeli koja zajedno s konstrukcijom grabi ka vrhu nalazi se i jedan računar. On takođe budno u stopu motri na kupolin put, a s aspekta bezbednosti ima ni malo zanemarljiv zadatak da upozori, a zatim i zaustavi hidraulični sistem za dizanje ukoliko dođe do nekih nepredviđenih situacija.

Celovita predstava

Reč je o sistemu za akviziciju podataka i upravljanja ED 1000 LAB koji je proizvođač — privatna beogradska firma „Elektronik dizajn" ustupila graditeljima Hrama da bi bezbednost ovog poduhvata bila potpuna.

— Sistem ima 48 analognih ulaznih i 32 ulazno izlazna digitalna kanala — objašnjava Slobodan Kotri, vlasnik „Elektronik dizajna". — Za ovu namenu programiran je tako da meri pritiske u cilindrima svake od ukupno 16 hidrauličnih presa koje podižu kupolu, zatim pređeni put, napon u betonu i armaturama kao i nagib. Sve ove podatke prikazuje u realnom vremenu grafički na monitoru.

Na komandnom pultu elektronsko-hidrauličnog sistema za dizanje registruju se međutim samo naponi i hodovi kupole. Kada se sa ovog mesta pritiskom na dugme izda komanda da kupola krene gore (dinamika dizanja je takva da konstrukcija u jednom cugu pređe 300 milimetara), na displeju pulta se pojavljuje poviaine informacije: digitalni podatak o naponu konstrukcije i postignutoj visini.

ED 1000 sistem za razliku od ovog komandnog pulta koji prikazuje trenutnu situaciju stanja tokom podizanja registruje grafički u vremenskom kontinuitetu. Zahvaljujući tome u svakom trenutku može se steći celovita predstava o svim ostvarenim fazama dizanja. Osim toga, svi ovi podaci beleže se na disketi, što omogućava njihovu kasniju analizu i obradu. Kažu da će ti zapisi sa disketa biti prvi majdan podataka za brojne buduće magistarske i doktorske radove. Uz to, biće to i prvi način da se jedan ovakav graditeljski poduhvat dokumentovano sačuva za budućnost.


Foto: Predrag Todorović

Naročito važna primena ED 1000 sistema na svetosavskom gradilištu je merenje nagiba kupole. Ovaj podatak graditeljima praktično obezbeđuje samo „Elektronik dizajn". Računar prikuplja podatke pomoću četiri elektronske libela, odnosno četiri analogna potenciometra postavljena na četiri mesta na periferiji kupole. Ovi uređaji su međusobno povezani crevima ispunjenim vodom. U zavisnosti od nagiba konstrukcije po sistemu spojenih sudova potenciometri menjaju vrednost, a računar sve to u realnom vremenu na monitoru registruje analogno i grafički. Podaci se izbacuju na svaka dve sekunde, tako da graditelji bukvalno u svakom trenutku znaju kako stvari stoje.

Kupola leti u nebo

Do neravnomernog podizanja kupole može doći zbog toga što oslonci — čelične kulije na kojima kupola leži, usled velike težine konstrukcije mogu da potonu, što je uobičajena pojava pri ovakvim operacijama. Bez obzira na sve brižne proračune, nemoguće je unapred predvideti koliko će i kada jedan od četiri oslonca kupole potonuti. Ukoliko računar registruje nagib veći od predviđenog (a dozvoljeno odstupanje je jedan milimetar) — uključuje se alarm i slede automatske naredbe za isključenje sistema. Tada se rad presa na mestu koje je „skrivilo" prekomerni nagib koriguje ručno, a zatim hidraulika nastavlja automatski rad.

Računar će upozoriti i ukoliko dođe do nepredviđenih promena u pritisku hidrauličnih presa. Svaki od 16 hidrauličnih cilindara koji podižu kupolu može da razvije pritisak do 350 bara i silu od tri i po hiljade kilonjutna u automatskom režimu. U ručnom režimu svaki cilindar može da ostvari duplo veću silu pri hodu od 300 milimetara. Prese kao izvor snage koriste četiri elektromotora od po 15 kilovata i osam klipno-radijalnih hidrauličnih pumpi. Kupola osvaja vrh step-by step, a posle svakog koraka pod nju se ugrađuju betonske ploče koje je podupiru. Kada stigne do četrdesetog metra biće postavljena na četiri noseća luka, a zatim će biti podignuti i pandatifi — prelazna forma od četvrtastog ka sfernom obliku.

Podizanje ovakve konstrukcije delikatan je posao i stoga što je sastavljena od mnoštva betonskih elemenata koji su u celinu montirani na zemlji. ED 1000 sistem meri zato i napone u presecima konstrukcije da bi se u svakom trenutku pratilo ponašanje materijala prilikom podizanja.

Softverski paket za ovu namenu napravili su programeri beogradskog IMR. Sistem ED 1000 ustupljen je KMG „Trudbeniku" za ovu priliku besplatno, a inače staje između pet i šest hiljada dolara. Cena ovakvog sistema za akviziciju podataka i upravljanje kreće se inače od dve do dvadeset hiljada dolara, zavisno od konfiguracije. Na gradilištu Hrama Svetog Save, međutim, nije premijerno upotrebljen u građevinarstvu. Jedan je već korišćen na beogradskom građevinskom fakultetu za beleženje podataka prilikom proba s veštačkim zemljotresom.

Fleksibilni hardver

Sistem ED 1000 međutim može da posluži i za mnoge druge aplikacije — na primer u robotici, automatskom testiranju, procesnom upravljanju u industriji, instrumentaciji... Slobodan Kotri naročito napominje da je reč o sistemu koji je kompatibilan sa industrijskim standardom IBM PC, kao i sa najkvalitetnijim softverskim paketima za akviziciju podataka i upravljanje. Modularni koncept sistema omogućava korisnicima velike uštede, jer korisnik sam specificira broj i tip I/O. Time se zapravo investira samo u ona komponente koje su potrebne da bi se dobila željena konfiguracija.

— Sistem za upravljanje 16 peći u jednoj fabrici, na primer, može se realizovati sa jednim I/O modulom. Pošto ova primena ne zahteva brojač vremensko kolo i analogne izlaze, korisnik ih ne mora nabavljati, što nije slučaj sa fiksnim hardverom za koji je obavezna nabavka i onih modula koji korisniku ne trebaju. Ukoliko se početni zahtevi konfiguracije ED 1000 sistema kasnije promene, korisnik ga može prilagoditi dodavanjem novih I/O modula, a eventualni višak već nabavljenih upotrebiti za formiranje novih sistema, objašnjava Kotri. Sisteme poput ED 1000 proizvode na primer „Bur — Brown" serija PCI-2000 i „Keithley" model 500.

Hardver ED 1000 sistema sastoji se od interfejsnih modula ED 1001IF i ED1015PCIF, ulazno-izlaznog modula ED 1000BUS modula ED1900PSU sistema za napajanje i konektora, kablova i pribora. Ovaj sistem radi s preko 20 programskih paketa podeljenih u tri grupe. U prvu spadaju programi upravljani menijem poput LABTECH NOTEBOOK — Integrated Data Acquisition Control and Analysis Software, LABTECH REAL TIME ACCESS — Software for Real Time Data Analysis and Control, RELAY LADDER LOGIC — Software for Process Monitoring and Control i drugi. U sledeću grupu spadaju softverski drajveri za jezike opšte namene koji omogućavaju pisanje programa za akviziciju podataka i upravljanje u bejziku, C-u i turbo paskalu. U treću grupu spadaju programi za specijalne aplikacije — ASYST, LOTUS i PEDAS.

U „Elektronik dizajnu" kažu da sistem ED 1000 ima takvu softversku podršku i „familijarnost" sa PC računarima da ga u svojim aplikacijama mogu koristiti i oni koji nisu računarski obrazovani.

Atari MEGA ST

Mega po mega pogača

Atari i dalje neumorno izbacuje nove modele. Dok se ljubitelji PC znaka još uvek zlopate sa segmentiranom memorijom, ST je već uveliko zakoračio u MEGA eru.

Marko Kirić

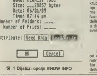
U doba pojavljivanja prvih računara ST serije na tržištu IBM se sa svojim PC modelima već čvrsto ukorenio na poslovnom delu tržišta. Odlučujuću ulogu su pri tome odigrale in fakto renomirano ime, već poslovičan kvalitet i činjenice da se radilo o prvom prodoru mikroračunara u taj deo tržišta. Ubrzo zatim se pojavio i trend 16-bitnih računara koji je naizvesno sledio i „veliki plavi", ali zbog kompatibilnosti sa ranijim modelima novi Čarlijevi modeli su vukli i teret 8-bitnog nasleđa. Prvi štali 16 bitni računar koji se pojavio i opstao na tržištu je bio naravno „makintoš" sagrađen oko Motorole 68000 koja se pokazala kao zaista savremen, moćan i brz procesor. Takav procesor je prosto zahtevao pisanje novog izuzetno ljubaznog operativnog sistema koji se vrlo brzo proslavio i postao omiljen u onoj klasi korisnika koji pre svega žele da u svakom trenutku vide šta rade i koji žele da računar koriste odmah po kupovini bez mnogo zamajavanja učenjem komandi.

Pored takva dva giganta firme koje je ranije bile poznata isključivo po igrama (na kraju tako je i odbila ime) nije imala nikakvog izgleda na tržištu ukoliko bi sledila njihovu filozofiju po kojoj je veličina cifre direktan odraz kvaliteta. Formula za uspeh Džeka Tremiela je bila koliko jednostavna, toliko i dobra: *Power without the price*. Novi računari su imali čistu 16/32 arhitekturu, izvanredan monitor, koristili su savremenije 3.5" diskete, a operativni sistem je od te mere bio nalik „makintoševom" da su naprlike sa Jabukom bile prosto neizbežne. Stvari se izgladila tako što je Atarijev (odnosno DR-ovj GEM pretrpeo minimalne, uglavnom kozmetičke izmene. Novi računari su imali na raspolaganju istu količinu memorije kao i „stariji brat" i približno iste mogućnosti, ali je zato cena bila više nego dvostruko niža!

Najlepša stvar u svemu tome je činjenica da je niska cena postignuta štednjom na kozmetici odnosno kućištu, pažljivim izborom isporučilaca delova i na žalost nešto slabijom tastaturom, ali zato brzina i kvalitet nisu ništa od ovoga osetili. Tako je put prema onom delu tržišta koji pokriva veoma veliku i raznoliku klasu kućnih korisnika i hakere bio širom otvoren, što se odrazilo i na prodaji — tokom sledeće dve godine ST modeli su, naročito u Evropi, postali pravi hit. Posle igre sa mnogobrojnim varijacijama dva osnovna modela u Atariju su shvatili da nije pametno oslanjati se samo na ovo prilično ćudljivo tržište i povukli su još jedan pametan potez: izbacili su na tržište MEGA ST seriju.

Cilj ovog manevra je jasan — svim dobrim osobinama serije dodati kvalitetnu tastaturu (na koju je inače bilo najviše zamerki), omogućiti memorijsku raskoš i sve zatvoriti u solidno kućište. Tako koncipirana MEGA serija cilja pre svega na studente i univerzitetske ustanove za šta su odbar primer predstavljali BBC i Apple II. Ovakav potez nije bio bez osnove jer je koncepcija ovih računara gotovo idealna za takve svrhe — to se pre svega odnosi na brzinu, cenu i količinu memorije. Pored toga drugi cilj je bile obiseti primene koje doživljava eksplozivni rast poslednjih godina — stono izdavaštvo. Iako je Makintoš na ovom polju još uvek suveren, u Atariju sve nade polažu u veliku grupu malih firmi kojima se ne isplati nabavka jednog takvog sistema, ali po Atarijevim cenama računica postaje sasvim drugačija.

Na tržištu se trenutno mogu naći tri MEGA ST modela: MEGA ST1, MEGA ST2 i MEGA ST4, koji se razlikuju samo po količini ugrađene memorije. Prvobitno su se pojavili samo MEGA ST2 i MEGA ST4, a krajem prošle godine i MEGA ST1. Stiče se utisak da ovaj model nije bio planiran, ali da su tržišne prilike učinile svoje. Naime, kad se potencijalni kupac sa ograničenom svotom novca (a to je upravo Atarijeva ciljna grupa) odlučuje za kupovinu prvog računara MEGA ST2 sa svojom cenom od oko 2500 DEM bez obzira na šta je nače imali mnogo šansi protiv modela 1040 ST, ali zato MEGA ST1, skuplji od 1040 ST za nekih 500 DEM, u novom kućištu, sa novom tastaturom, novom verzijom operativnog sistema, ugrađenim bliter čipom i mogućnošću jednostavnog dograđivanja na jedan od jačih modela ima veoma dobre izglede.



Sl. 1 Dijalozi opcije SHOW INFO

Razlike

Iako su potpuno kompatibilni sa starom serijom MEGA modeli nisu potpuno isti. Pored čisto kozmetičkih izvršen je i čitav niz sitnijih i krupnijih izmena čiji je krajnji rezultat eliminisanje skoro svih postojećih mana stare serije. Ona je dobijen izuzetno ozbiljan i upotrebljiv računar. Uneta izmene i osvaženja se mogu podeliti na kozmetičke, hardverske i softverske. U daljem tekstu ćemo se posebno osvrnuti na svaku grupu.

Kozmetika

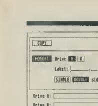
Prvo što upada u oči pri pogledu na neki od novih modela je novo, moderno i funkcionalno kućište i odvojena tastatura. Samo kućište je slim-line tipa, sa ugrađenom dvostranom disk jedinicom na desnoj strani prednje ploče i svim potrebnim konektorima na zadnjoj ploči. Time je konačno eliminisana „telefonska centrala" na stolu, rešeno pitanje mesta za monitor i postignut izgled koji je u skladu sa mogućnostima mašine. Na poleđini se nalaze svi standardni priključci: ulaz za napajanje, izlazi za drugu disk jedinicu i monitor, RS 232, CENTRONICS, MIDI IN, MIDI OUT, DMA port, glavni prekidač i reset taster. Pored toga tu su još i mali i veoma tihi ventilator i otvor koji omogućava direktan pristup na CPU za dalja proširenja.

Tastatura

Tastatura je smeštena u novo, zasebno kućište na kome se nalaze i portovi za miša i džojstik, a sa centralnom jedinicom je povezuje kabl po američkom PTT standardu. Kućište je tanko i veoma lepo leži pod rukom, a za one koji vole nagnutu tastaturu postoje dve nožica kojima se može regulisati nagib. Portovi za miša i džojstik su smešteni na donjoj strani kućišta tastature a kablovi su obezbeđeni od savijanja posebnim vodicama. Sama tastatura je na prvi pogled identična onoj kod starih modela, ali već posle nekoliko otkucanih reči postaje jasno da su tasteri bitno poboljšani. Gumeni čepići koji su na ranijim modelima bili smešteni ispod svakog tastera sada su zamenjeni klasičnim tasterima sa oprugama, a kapice su ostale iste, što znači nešto šire od klasičnih tastera. Najčešći izrazi koje su korisnici koristili za opisivanje stare tastature su bili „mekano", „mlitavo" pa čak i „gnjecavo" ili „ljigavo" — sve te primedbe sada otpadaju jer nova tastatura pruža izuzetno izbalansiran otpor pri pritisku, tako da se ne odbija utisak da prst propada pri pritisku, ali istovremeno nije potrebno ni boriti se sa tasterima (to se naročito odnosi na tastere koji se kucaju malim prstom). Što se širine tastera tiče, mnogi korisnici bi voleli da vide tastaturu sa užim kapicama, ali to je ipak stvar ukusa — ovakva kakva je, širina kapica odgovara korisnicima koji kucaju ne pomerajući ruke, već samo prste.

Hardverske izmene

Prva i sa gledišta korisnika najprimetnije novost je ugradnja bliter-čipa. Bliter ili *Bit Block Transfer Processor* je u suštini mikroprocesor čiji osnovni zadatak je vrlo brzo premeštanje blokova memorije sa jedne memorijske lokacije na drugu. To se postiže tako što kad se pojavi zahtev za takvim prenosom, glavni mikroprocesor po izvršavanju tekuće naredbe predaje kontrolu nad magistralom bliteru, koji je po završenom prenosu vraća glavnom procesoru. Princip je sličan DMA prenosu, samo što se ovde prenos obavlja isključivo unutar memorije, dakle bez angažovanja I/O kontrolera. Prednosti ovakvog prenosa su najočiglednije pri radu sa grafikom (naročito animacije i CAD primena) a zatim i na nekim osnovnim grafičkim elementima samog operativnog sistema (skrolovanje ekrana, obnavljanje i pre-



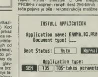
Sl 2 Dijalog za formatiranje

meštanje prozora kao i sve VDI rutine prikazivanje raznih vrsta teksta na ekranu itd) Kod svih grafičkih operacija važi pravilo da je prikaz utoliko brži i čistiji ukoliko je brzina prenosa veća Medutim tu se krije i jedna zamka korišćenje bitera za pomeranje blokova manjih od jednog Kb ne samo da pokazuje željeni efekat, već u nekim slučajevima može doći i do primetnog usporenja! Razlog leži u činjenici da pri prenosu tako malih delova memorije glavnom mikroprocesoru koji je i sam veoma brz treba mnogo manje vremena za odvijanje te kve sitnice nego za predavanje kontrole biteru i čekanje da ovaj obavi zadatak i vrati kontrolu Zbog toga je i predviđena mogućnost da se biter isključi iz desktopa Bliter sada spada u standardnu opremu MEGA serije ali pri kupovini nekog od ovih modela ne bi bilo naodmet proveriti da li je u konkretnom primerku zaista i ugrađen Naime događa se (naročito u Nemačkoj) da prodavac bude „zaboravan" i na taj način pokuša da ostvari ekstra zaradu Da li je bliter ugrađen ili ne može se najlakše proveriti tako što se računar uključi i povuče meni **EXTRAS (OPTIONS)** Ukoliko je bliter ugrađen na dnu menija će se pojaviti opcija **BLITTER** koja je od ostalih odvojena isprekidanom si vom linijom Ako bliter **nije ugrađen,** ova opcija se neće pojaviti

Sledeća novost je ugrađen časovnik sa kalendarom sa sopstvenim napajanjem Ovo ni u kom slučaju ne bi trebalo da predstavlja novost jer je tako samo ispravljen jedan veliki propust konstruktora Atarija Časovnik bez svog napajanja koji je bio ugrađivan u starije modele su samo najupornijі AtariSTi bili spremni da podešavaju prilikom svakog uključenja računara i samo se po sebi razume da jedan računar koji po svojim performansama i mogućnostima pretenduje na profesionalnu primenu mora da ima ugrađeno napajanje takvog časovnika. Časovnik koji se ugrađuje u MEGA modele je rešen drugačije nego kod starih modela znatno je precizniji a o njegovom napajanju u vreme kada je računar isključen se brinu dva NiCd akumulatora koji se pune kad računar radi

Port za proširenja je ono što bi se mogio nazvati najkrupnijim zaokretom u dosadašnjoj koncepciji Stariji ST modeli su predstavljali zatvoren sistem što sa mikroprocesorom ove klase takođe ne može da se nazove pametnim rešenjem To je je još jedna ilustrativna činjenica koja pokazuje ko i u kojoj meri je bio uzor pri stvaranju ove serije Kao što su i u Apple-u shvatili negativan uticaj ovakve koncepcije na prodaju i ispravili kod svog najnovijeg modela Meke tako su i Atarijevi konstruktori najzad omogućili svom glavnom adutu (M68000) disanje „punim plućima" Ovaj port čini 64-polni konektor koji preslikava kompletnu I/O (izlazno-ulaznu) magistralu Motorole M68000 Time je najzad otvorena mogućnost priključivanja matematičkih koprocesora (na raspolaganju je kartica sa M68881 na 20 MHz) raznih kontrolnih i mernih uređaja Na kraju krajeva Motorolina familija procesora 680×0 polako ali sigurno stiče apsolutni primat u svetu kontrolnih uređaja i postaje ono što je nekad bio Z80 u raznim varijantama Kad već tako stoje stvari, zaista bi bilo krajnje inertno i kratkovido ne omogućiti ovakvom računaru priključivanje na takve uređaje tim pre što je ciljna grupa korisnika upravo ona koja ima najviše zahteva za takvom mogućnošću

Sledeća hardverska izmena koju bi svakako trebalo prokomentarisati je uvođenje megabitnih memorijskih čipova i po tome je serija i dobila ime (MEGA se odnosi na MEGAbitne čipove a ne na MEGA memoriju) Čak je operativni sistem smešten u dve megabitne PROM-a naspram ranijih šest 256-bitnih Prateća pojava je bila i rekonstrukcija matične ploče koja sad sadrži 35 standardnih integralnih kola dok broj memorijskih kola zavisi od modela (32 u MEGA ST4 i 16 u MEGA ST2) Naravno štampana ploča je kompletno redizajnirana u skladu sa uvedenim izmenama i promenjenim dimenzijama kućišta



Sl 3 Dijalog za instaliranje

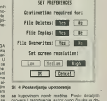
Softverske izmene

Verzija TOS-a koja je ugrađivana u MEGA mašine do kraja prošle godine je nosila oznaku 1 2 a od početka ove godine pojavila se i verzija 1 4 (stara verzija je imala oznaku 0 19) U odnosu na staru verziju u novim verzijama se može primetiti čitav niz poboljšanja i dopuna Ono što se vidi na prvi pogled je izmenjeni desktop sa izmenjenim i poboljšanim opcijama za formatiranje informacija, podešavanje sistemskih upozorenja i naravno, znatno boljim i bržim otvaranjem zatvaranjem odnevljanjem i skrolovanjem prozora Skrolovanje prozora se sada može vršiti i kontinualno odnosno ako se taster miša drži pritisnut na strelici ili polju za skrolovanje dolazi do ponavljanja što važi i za sve GEM programe Opcije **SAVE DESKTOP** i **PRINT SCREEN** su dobile svoje dijaloge za potvrdu a prikazivanje sadržaja ASCII fajla na ekranu se može ostvariti i pritiskom na <Return> Pored toga, opcija za formatiranje disketa je sada znatno fleksibilnija i komfornija za upotrebu a isto se odnosi i na opciju **SHOW INFO** Pri kopiranju disketa iz sistema, kao i pri štampanju fajlova sa deska pada u oči da se ceo proces odvija sa manje odrećanja izvornoj disketi (pri kopiranju je potrebno manje izmena disketa pri radu sa jednim drajvom) što navodi na zaključak da je odgovarajući bafer značajno povećan To bi bila najkrupnija promena sa stanovišta korisnika Najvažnije suštinske izmene i dopune su

— Memorijski prostor za GEM je povećan sa 6 na 16 kB što drugim rečima znači da je već čuveni problem sa 40 foldera sada eliminisan što će naročito obradovati vlasnike hard diskova.
— CRC_ERROR koji šalje disk-jedinica se sada uzima u obzir za razliku od starijih verzija TOS-a a šta to znači u praksi svima je dobro poznato
— Rutina za kontrolu RS 232 interfejsa je potpuno nova Naime stara verzija je bila toliko puna bagova da se jednostavno više isplatilo napisati potpuno novu rutinu nego ispravljati staru
— Dodate su rutine za kontrolu blitera i sistemskog časovnika
— Proširen je TOS heder
— GEM u su dodate rutine **appl_trecord** i **appl_tplay**

Pošto je ROM i dalje veličine 192 Kb da bi se obezbedilo mesto za nove rutine trebalo je izvršiti nekoliko zahvata Tako je izvršena optimizacija u odnosu na dužinu i brzinu reprogramirani su BIOS Screen Manager Line-A a samim tim i VDI a takođe i rutina za štampanje ekrana i par BIOS rutina

Po pojavi novih mašina a u vezi softverskih izmena pojavile su se glasine da novi TOS **ne dozvoljava pristup trakama iznad 80,** što bi u praksi značilo da nije moguće koristiti format veći od 800 kB!! Naravno pokazalo se da su ipak u pitanju samo glasine — novi TOS savršeno podržava sve formate koje omogućava ugrađeni drajv (provereno!) i da su dezinformacije ove vrste potekle od onih korisnika čiji drajvovi nisu bili sposobni za ostvarivanje većih formata. Posledica takvih glasina, koje su se uzgred budi rečeno pojavile čak i u jednom našem kompjuterskom časopisu je logično bila da i neki kvalitetni i ozbiljni programi ne rade na novom TOS-u što je u priličnoj meri uznemirilo korisnike i navelo ih na oklevanje



Sl 4 Postavljanje upozorenja

sa kupovinom novih mašina Posle detaljnih provera i raspitivanja autor ovog članka je došao do zaključka da **svi ozbiljni programi rade bez ikakvih problema** na novim mašinama Jedini izuzetak predstavlja stara verzija TEMPUS-a ali taj program je vrlo često pravio probleme i na raznim primercima mašina sa starim TOS-om što znači da je u pitanju sam program i način njegovog pisanja (poznato je da je organizacija ovog programa vrlo specifična i neregularna) Pored toga **neregularno napisane hakerske igre** takođe imaju šanse da ne rade na novom TOS-u a glavni razlog je nepridržavanje nekih normi i preporuka proizvođača programske opreme što je uostalom i glavna karakteristika one klase programera koje u negativnom kontekstu nazivamo hakerima

Opšti utisak

Rad sa MEGA mašinama bi se u poređenju sa starim modelima ST serije kao i sa drugim modernim 16-bitnim računarima mogao



Sl 5 Nova upozorenja

nazvati vrlo udobnim i prijatnim. To je tipičan primer univerzalne *user friendly* nastrojene mašine velike snage i velikih mogućnosti i prava šteta je što se ova serija nije pojavila par godina ranije. No i ovako možemo slobodno da kažemo da je pojavom novih modela ST serija konačno preboleła sve dečije bolesti i definitivno izborila svoje mesto pod suncem na vrlo širokom i ćudljivom tržištu personalnih kompjutera.

Radi sticanja potpunije slike treba još reći i to da GEM okruženje koje ova serija koristi nalazi na oduševljen prijem kod svih onih koji teže vizuelnoj predstavi rada, a u isto vreme i prilično odbojan, a često i podsmešljiv stav korisnika naviknutih na MS DOS i slične sisteme što na prvi pogled deluje nelogično, ali činjenica je da niko prema ovom sistemu ne ostaje ravnodušan, a za koji sistem će se korisnik opredeliti u krajnjoj liniji predstavlja stvar ukusa i ličnog afiniteta. Činjenica je takođe da ova serija nema velikih izgleda za uspeh na čisto poslovnom delu tržišta, ali zato pušta sve dublje korene u univerzitetskim i istraživačkim organi-



zacijama, naročito u oblasti fundamentalnih nauka — da pomenemo samo Univerzitet u Cirihu, a od naših ustanova prirodno-matematički fakultet u Beogradu i Institut za fiziku u Zemunu. U takvim sredinama ova serija ima sve uslove da, pored sjajnih mogućnosti u oblasti DTP-a (stonog izdavaštva), pokaže svu svoju snagu na rešavanju veoma kompleksnih zadataka u kojima se vrlo često barata blokovima podataka reda veličine megabajta.

Na planu cena, poznati Atarijev slogan pod kojim je lansirana ST serija, *Power Without the Price*, gubi nešto od svoje ubedljivosti, ali ipak stoji činjenica da jedan ovako koncipiran kompletan sistam po karakteristikama može sasvim ravnopravno da se poredi sa odgovarajućom AT konfiguracijom, dok poređenje cene izmamljuje blage osmehe. Najkraće rečeno, ovo je računar sa ubedljivo najpovoljnijim odnosom **cena/kvalitet**, uzimajući u obzir i broj i kvalitet dostupne programske opreme.

ST_NEWS u „Računarima"

Disk — magazin za atariste

Ako već imate računar visokih grafičkih mogućnosti pravi je greh ne iskoristiti ga za — izdavanje magazina na disku. Od ovog broja „Računari" vam ekskluzivno nude najbolji evropski i svetski disk-magazin za korisnike Atari ST računara.

Računarski magazin ST NEWS izdaju dvojica Holanđana od kojih je prvi, Stefan Posthuma, zadužen za grafičku i tehničku opremu magazina, te za izbor tekstova koji redovno pristižu iz Evrope. Drugi član redakcije je i našim korisnicima poznat Richard Karsmakers, autor čuvenih VDU programa koji s lakoćom isteruju sve viruse iz ST računara. On je zadužen za virusologiju, sveže vesti, top listu igara itd.

Magazin izlazi 6 do 7 puta godišnje, i to u prvoj polovini godine ređe, a u drugoj češće. Svaki broj daje oko pola megabajta raznih tekstova i podataka počev od prikaza igara pa do ubijanja virusa.

Trenutno se distribuira na četiri kontinenta i dvadeset zemalja. U Africi je samo jedna zemlja, Južna Afrika. U Americi su to U.S.A. i Kanada, a u Okeaniji Australija i Novi Zeland. U Evropi je potpuno pokrivena Skandinavija (Finska, Danska, Norveška, Švedska) i Mediteran (Španija, Francuska, Italija, Jugoslavija, Grčka, Turska) dok se u srednjoj Evropi distribucija vrši u Austriji, Švajcarskoj, Zapadnoj Nemačkoj, Belgiji i naravno Holandiji, a postoji i distributer u Engleskoj. Sve u svemu može se reći da se radi o magazinu koji ima ubedljivo najveću površinu prostiranja, a time i najveći „tiraž" sa kojim se možda može meriti samo par američkih računarskih magazina.

Magazin je naravno isključivo na engleskom jeziku kao univerzalnom jeziku programera i korisnika računara. Holanđani su nama, za razliku od Nemaca i Francuza, poznati kao izrazito kosmopolitski orijentisani, pa je i to jedan od razloga za ovako veliku rasprostranjenost magazina. Za sada jedino nema distribucije u zemljama istočnog bloka, što ne znači da se magazin i tamo ne čita.

Pogled iznutra

Srce magazina je program ST NEWS.PRG koji po startovanju daje klasičnu liniju menija, a svaki meni sadrži naslove tekstova. Taj program Stefan stalno dorađuje, menja, poboljšava itd. Trenutno je dugačak nekih četrdesetak kilobajta, što je prilično mnogo kad se uzme u obzir da se radi o programu kompletno pisanom na mašinskom jeziku. U njemu se između ostalog nalaze izuzetno dobre rutine za skrolovanje teksta, odličan program za sintetizovanje zvuka itd. Tekstovi se nalaze u jednom ogromnom fajlu od nekih 300 K. Svi tekstovi su u WP modu pa ih iz samog programa, koji služi kao interaktivna korisnička okolina za biranje i čitanje teksta, možete snimiti na disk tako da ih 1st Word može direktno učitati. Uz svaki broj dobijate i muziku koju možete slušati za vreme čitanja tekstova. To je ujedno i jedini razlog za uključivanje softvera za sintezu zvuka u program. Softver za sintezu zvuka se isto tako stalno dorađuje kako bi maksimalno iskoristio mogućnosti ST-ovog zvučnog čipa, no time se bave specijalizovani spoljni saradnici.

Postoji pet menija. Prvi (Features) donosi rešenja par igara, razne tekuće zanimljivosti kao i jednu sliku koju doduše ne možete direktno snimiti, ali zato postoje snapshoot programi. Tu je zatim tekući virusološki izveštaj (najnovija vest — u USA postoji virus koji može fizički uništiti tvrdi disk) te izveštaj sa nekog putovanja. Ričard i Stefan gotovo redovno idu u posetu raznim evropskim hakerima i sve što im se usput događa zapisuju u jedan portabilni računar.

Drugi meni (Regulars), kao što mu samo ime kaže, sadrži redovne rubrike u koje spada i lista svih distributera, kontakt adrese, top lista igara, uvodnik i mnogo softvereskih vrućih vesti od kojih je 90% na žalost posvećeno igrama. Stefan je inače po vokaciji haker pa se to odražava na koncepciju magazina koji je gotovo 50% posvećen igrama.

U trećem meniju (Education) se nalaze gotovo čisti programski članci. Obično ih je tri do pet, posvećenih mahom radu na mašinskom jeziku, a među njih se šverca i izvesna redovna rubrika koja sadrži po nekoliko stranica teksta manje više šašavog sadržaja koji nema nikakve veze sa programiranjem a ni sa životom.

U četvrtom meniju (Reviews) su prikazi. Približno deset prikaza igara, jedne do dve knjige, dva do četiri prava aplikaciona programa i neki hardver često iz underground-pogona.

Poslednji meni (Options) je pomoćni i omogućava invertovanje ekrana, uključivanje i isključivanje muzike, prikazivanje Help-ekrana te posebnu opciju sa programerskim čudesima na delu. U poslednjem broju se tu nalazio tekst koji se stalno skroluje kao jedna linija s leva na desno. Posle izvesnog vremena linija postaje zakošena, zatim se lagano njiše da bi završila kao sinusoida i sve to uz potpuno glatko pomeranje slova i glatku promenu oblika linije.

Računari u ST NEWS

Počev od ovog broja preko HELP-a možete dobiti i ST NEWS. Distribucija počinje od najnovijeg broja (4.2) a ako se pokaže interesovanje biće nabavljani i raniji brojevi (poslednje godište je sasvim sigurno vrlo zanimljivo).

Ukoliko mislite da možete napisati nešto zanimljivo i želite da to pročitaju ljudi na četiri kontinenta, vrata su vam svakako otvorena. ST NEWS je potpuno otvoren magazin koji rado prima sve dobre tekstove odakle god dolazili. Neophodno je naravno da tekst bude na engleskom, a ukoliko se radi o rutini ili programu da bude solidno dokumentovan sa propratnim tekstom u kome je detaljno objašnjeno šta, kako i zašto se događa u programu. Priloge možete poslati HELP-u odakle će svakih 30 do 60 dana biti slani u Holandiju. Istim kanalom će odlaziti i domaći PD programi koji budu imali sve poruke i pristojno uputstvo na engleskom jeziku.

Žarko Berberski

Rajko Dujmić — kompozitor, muzičar, producent

Softver u glavi

Rajko Dujmić — trenutno najpoznatiji kao kompozitor sada već slavne pobedničke kompozicije „Rock me, baby" sa „Pesme Evrovizije". Savršeni violinista, sjajni klavijaturista, kompozitor, aranžer, producent, vođa „Novih fosila", čovek kome je život pesma (ah, kako to lepo zvuči), a vrtoglavi tiraži, pobede i nagrade prirodni ritam, u ovako dizajniranom prostoru jeste u svom prirodnom elementu. Razgovor smo vodili u „Studiju Lisinski", gde je smešten prvi jugoslovenski višekanalni tonski studio za digitalno snimanje, zgodno ušuškan 2 m ispod zemlje.

Vesna Ćosić

„Studio Lisinski" nalazi se ispod koncertne dvorane „Vatroslav Lisinski". Don Dejvis (Davis) iz američke firme „Synergetic Audio Concepts" razradio je tzv. LEDE (Live End Dead End) koncepciju po kojoj je obrađena i režija „Studija Lisinski". Ispunjena velikim studijskim računarom, tonskim pultom „Solid State Logic", analognim i digitalnim magnetofonima, te uređajima za procesiranje zvuka poput kompresora, ekspandera, digitalnih reverberatora, uređajima za kašnjenje, harmonizerima, ekvilajzerima i ostalim čudesima za podršku muzici,

• I dalje je najbitnije sviranje. Kompjuter ne znači da će on svirati umjesto nas.

ova čarobna kutija je prirodno okruženje za hit mejkere od formata. U društvu drugara iz kraja, ležeran i inventivan, Rajko Dujmić je šarmantan sagovornik koji razgovor o notama ne vodi „na note". Njegovi saradnici Dragan Čačinović i Zdravko Kunić, sjajni momci iz „Studija Lisinski", primer su timskog sklada i druželjubive spontanosti.

***Računari:** Priznajem, u dilemi sam da li je ovaj studio hakersko gnezdo ili laboratorija za note? Producent ste na glasu pa vas pitam je li to proširena upotreba tehnike ili proširena upotreba muzike?*

Rajko Dujmić: (smeje se) Kako se uzme. Trenutno smo na početku miksa. Snimljeni materijal stavljamo na pult. Pametan momak je ovaj pultić i sve zapamti. Ima kompjuter koji nas zapravo oslobađa svih fizičkih radnji u finalnoj fazi. Mi više, recimo, ne vučemo regiere nego se oni sami vozaju po zapamćenoj melodici. Ovo je kraj prvog dana miksa. Znači, imamo još jedno osam dana.

***Računari:** Kakav računar stoji iza svega?*

Rajko Dujmić: To je studijski računar „Solid State" koji se brine o tome da čovjek ne mora imati puno papira oko sebe nego se svi podaci upisuju u kompjuter, počev od naziva pjesme do loga gdje počinje refren, gdje počinje bilo šta. To se zove *queue punch*. Može se kompjuteru reći da pošalje magnetofon koji uopšte ne vidite na početak refrena, svirej *replay* ili, recimo, u nekakvoj pjesmi pjevač je jedan dio otpjevao preglasno, treba ga na tom mjestu stišati. Kompjuter to zapamti. Sada svaki sledeći put kada će se svirati ta pjesma on će je sam utišati na tom mjestu.

***Računari:** Kakva je softverska podrška?*

Rajko Dujmić: Dobije se zajedno sa kompjuterom na jednoj disketi. To je operativni sistem računara. Sa tim operativnim sistemom je obuhvaćena komunikacija pulta sa magnetofonom i sa korisnikom. Druga disketa se pravi tokom snimanja i ide zajedno sa trakom. Svaka traka ima svoju disketu. Tamo je zapisano na kojem kanalu na traci se šta nalazi, na kojoj je minutaži traka koja pjesma. Ne treba više gađati nego se kaže: Sviraj mi treću pjesmu. On će je sam naći.

***Računari:** Da li se sa ovim operativnim sistemom može koristiti i drugi softver? Recimo da se „prikače" na sintisajzer ovaj na računaju ku... Na „atariju" se može lepo svirati, zar ne?*

Rajko Dujmić: To se radi pomoću drugih kompjutera — jedan „atari" ... „Steinberg" i „Emulator-3", korišćen kao sempler kao izvorište zvuka. Cijele ploče programiraju se na „atariju". On je uspio sve instrumente programirati da sviraju sami. Čovjek je jedanput to odsvirao i on je to zapamtio, šta je trebalo ispraviti ispravilo se na njegovoj tastaturi i njegovim mišem bez sviranja. Onda se taj kompjuter brine o sviranju, a ovaj kompjuter da se to zapiše na tra-

• Mi smo ultraprofesionalna grupa i kompletno smo kao grupa ljudi koji žive od toga i dišu za to i ne damo se omesti. I sviramo za dvadeset kao i za dvije tisuće ljudi

ku. Mada međusobno nisu imali komunikaciju, što je velika šteta, tako da je ovaj kompjuter bio rob onome.

***Računari:** Ko je zapravo, kome bio rob?*

Rajko Dujmić: „Atari" je bio rob „Solid Stateu". Zato jer je po SMPT kodu, to je kod koji se zapiše na traku, on slušao zapravo magnetofon, a ne da magnetofon sluša njega.

***Računari:** A da li u principu mogu da se povežu?*

Rajko Dujmić: U principu mogu.

***Računari:** To znači da pravite muziku, a kompjuteri nisu povezani? Zašto? Da se kao u dobrom SF u ne udruže?*

Rajko Dujmić: (smeje se) Nemamo taj interfejs.

***Računari.** Ako sudimo po rezultatima, može i bez njega.*

Rajko Dujmić: Ovakav način rada zadovoljava skoro sve potrebe. Samo treba odlučiti što hoćete. Da li hoćete da se sa magnetofona određuje pozicija mjesta na kompjuter sekvenceru ili obratno. A za to obratno treba kupiti još jedan interfejs.

***Računari:** I onda bi „atari" sve „vozio". Programirate li?*

Rajko Dujmić: Programiranjem se bavi Duško Mandić. On je programer i ima i EMU 3. Iza „Syn klavira" i „Fermante" je treći u svjetskom rangu semplera i sekvancera. To je strahovičan instrument.

***Računari:** Priča se da Rajko Dujmić, čovek od uspeha, ima navođen softver za muziku. Da li je to istina?*

Rajko Dujmić: Pa, Dujmić ima najviše softvera u glavi, jer ja znam što hoću. Kada vas neko odvede u trgovinu sa igračkama ako ne znate koja vam se igračka sviđa onda se izgubite.

***Računari:** Da li to znači da se u vašem poslu stvari komplikuju kada osećate šta hoćete, ali to treba artikulisati saradnicima?*

Rajko Dujmić: Važno je, recimo, to da ja ne programiram. Jer to stravično puno energije gnta a i koncentracije, a meni je potrebna za taj uvjetno rečeno kreativni dio, mada je i ovo kreativni, naravno. Kada čovjek programira

• Volim pisati za žena. Ljepše mi je pisati za žene.

mora biti kreativac i nekim svojim sugestijama doprinijeti da to bude. Mi idemo zapravo na pultu smo prema humanizaciji ... svega.

***Računari:** Muzike?*

Rajko Dujmić: Muzike, muzike! Jer humaniziramo, radimo čak ponekad i namjerne greške. Namjerno negdje zavučemo na nazad da bi cijele pjesme ... da bi se dobio osjećaj ... *Lay-back* mi to zovemo, malo kao da se nagnemo na nazad. Kao što je sve stvar slobodnih asocijacija.

***Računari:** Kompjuter onda nije referenca za hit. Naprotiv. Bez „ljudskosti" to je savršenstvo sa manom.*

Rajko Dujmić. Malo ljudskosti uvijek damo. Jer računar može savršeno odsvirati u tempu i to se tako podesi da je to as. Međutim, to onda više nema taj šarm, a taj šarm se mora znati baš na kom mjestu šta, kako.

***Računari.** Strogo kontrolisana muzika?*

Rajko Dujmić: Pod kontrolom ... i naravno, mora se poznavati što sve nude aparati sa kojima radimo. Ako ne znam boje koje, recimo, ima Mandić u „Emulatoru-3", ja sam mrtav producent. Jer mi, recimo, ponudi dvadeset vrsta doboša ili klavira.

***Računari:** Užasno!*

Rajko Dujmić. I to dvadeset drastično različitih klavira! I naravno da ću se izgubiti ako ne znam koju boju hoću. A onda se mi dogovorimo uglavnom kod kuće, u stanu kod Duške ili kod bilo koga ... on je jedan od programera sa kojim najčešće radim, najbolje je opremljen. Ali ima i drugih ljudi sa kojima ja također sarađujem, koji su programeri, ljudi koji se time dave i kojima je to u malom prstu. Za to treba strašno puno vremena i jedan talenat i pare za opremu.

***Računari:** Ako je budućnost kompjutera u muzici u tome da se on iskoristi u onim segmentima gdje se može dobiti na vremenu i time po jeftiniti sama produkcija, zanima me koliko upotreba računara stvarno skraćuje vreme produkcije?*

Rajko Dujmić Pedeset posto. Duplo brže se radi ako se isprogramira kod kuće što mi generalno radimo na sve ali bazu postavimo. U studiju se ne programira. Skupo je. Sve je kod kuće isprogramirano. Bubanj, bas i ostale neke instrumente za koje znamo da moraju biti isprogramiramo kod kuće na „ataru" a onda ovde u studiju nasnimimo vokale koje ne možemo kod kuće raditi.

Računari *A šta recimo misle bubnjari klasični bubnjari o tih dvadeset vrsta bubnjeva na računaru? Znači li to da je obična svirka već stvar kamenog doba muzike?*

Rajko Dujmić Najsjajnija je kombinacija kada sviraju živi ljudi. Mi na žalost još nemamo takav muzički kompjuter — recimo „sinklavir" sviraju živi ljudi i onda se to samo triga računarom iz računara se uzima boja koja odgovara. A daleko najbolje se dobije kada trajgove snimate sa dobrim muzičarima. U Jugoslaviji recimo imate vrlo malo muzičara koji će zadovoljiti. Bubnjare konkretno ima ih na prste da izbrojimo. Recimo da njih uzmemo i da sa njihovim bubnjevima dođu ovdje i dok se postave mikrofoni i dok se nađe boja, toliko vremena se utroši da iako je rezultat bolji moramo gledati ekonomsku stranu. Boja je na jedan drugi način daje manju mogućnost izbora. Jer manja je mogućnost izbora u smislu produkcije i aranžiranja.

Računari *Kvinsi Džouns (Quincey Jones) promeni dvadeset bubnjara za jednu ploču!*

Rajko Dujmić I opet ih isto provlači kroz sinklavir i zapravo ih preuređuje na neki način.

Računari *Hm znači kompjuter nije rodno mesto hita samo „servis"?*

• Mi kompozitori smo zid iza riječi, na koji se riječi naslanjaju

Rajko Dujmić Oni koji su odvje radili iako nesu su na ljudski faktor sujetu kao problem. Međutim nema sve ovo pomaže da brzo dođemo do rezultata. Znači najbitnije je sviranje i dalje je najbitnije sviranje! Kompjuter ne znači da će on umjesto nas svirati.

Računari *Međutim neki kao da bez hardvera više ne mogu ni usta da otvore a kamoli da se prihvate instrumenta. Ne dešava li se muzika u stvari pored njih?*

Rajko Dujmić Ima cijelih produkcija u Jugoslaviji gdje kompjuter svira i to dijete koje poznaje tonove u oktavama zna. Osnovno je ipak odsvirati. Jeste da kompjuter puno pomaže ali ako nemaš šta da pomogneš on se neće sjetiti (smeje se)

Računari *Znači uloga kompjutera u svetskoj muzičkoj revoluciji je da pomogne da se što pre i na bolji način dođe do rezultata u onome što čovek zapravo hoće. I da živa svirka još nije prevaziđena!*

Rajko Dujmić Ma kakvi nikada i neće biti! Najbitnije je razmišljanje onih ljudi koji kupuju ploče. Ja se poistovjećujem sa ljudima koji slušaju muziku koju tog momenta radim.

Računari *Kako to izgleda poistovetiti se sa tim ljudima? Kako „teleportirati" jednu iluziju po meri mnogih?*

Rajko Dujmić Vrlo jednostavno. Stvar je imaginacije. Postavili se u to da si od šest do dva na poslu. I da si došao kući i da ručaš i upališ radio. I šta? Šta bi volio čuti tog časa? E onda počne znači treba napisati takvu pjesmu. Treba biti to dio tebe i svih dvadeset miliona ili dvadeset i dva koliko nas ima dio svačijeg života. Jer ako je nešto iracionalno nešto što se nije dogodilo ni meni i sigurno se nije dogodilo nikome od ljudi koji će to slušati i voljeti a osnovni cilj je da ono što mi proizvedemo i što će izaći — da se to svidi. To je mašta same do ljudi dolaze samo riječ i tonovi muzika.

Računari *kao kolektivni doživljaj i onda je to hit!*

Rajko Dujmić Muzika je najimaginarnija



Talentovani ultraprofesionalac Rajko Dujmić

umjetnost i jedina je u stanju jako promijeniti ljudsko raspoloženje. Ne možeš je pipati ni vidjeti ne možeš ju taknuti nijednim od čula koja su realnija od samoga sluha. Uho je najbliže mozgu mozak radi na slabu struju i na elektromagnetske valove a isto to je i ovo što čujemo iz zvučnika. Znači najbliža veza sa mozgom je preko zvuka. Ne kažu zabadava muziku kao terapiju u medicini. Na koncu čak i konjima u štalama puštaju muziku. I to se zna točno koju u koje doba u linijim štalama.

Računari *Kakav je recept za hit? Pomaže li računar da se lakše „umuti"?*

Rajko Dujmić Svejedno je. Hit je hit!

• Uloga kompjutera u svjetskoj muzičkoj revoluciji je da pomogne.

Računari *A ono vreme pre hita vreme pripreme i vreme oikoce šta bi to moglo biti?*

Rajko Dujmić Ja to ne radim sa računarom. Ja pišem pjesme za klavirom. Obično Hit se rodi.

Računari *iz glave za klavirom?*

Rajko Dujmić. Više iz duše. Kada krene iz glave onda je već vreme da se zatvara dućan.

Računari *Uh! To je komplikovano jer možemo se pitati gde je duša možda je nekom u glavi*

Rajko Dujmić To je već filozofija. Ne znam kad radim moje fraze i pjesme znaju zazvučati nelogično do zle boge ali su ugođene. Imam tu jednu atmosferu u kojoj recimo svaka riječ ima svoju zakonitost. Svaka riječ točno ima svoju intonaciju. Ne možete recimo „volim te" napraviti ili neku riječ koja se izgovara u padu sa silaznim akcentom. Recimo riječ „zdravo" ne možeš reći (peva) Zdravo zdravo (sa uzlaznim akcentom smeje se) — jer to nije to zar ne? Nego (peva) Zdravo! Znači već sama ta riječ zahtijeva da melodijska linija teče dolje u izgovoru. I tu smo mi spojeni mi kompozitori smo zid iza riječi na koji se riječi naslanjaju. Riječi su prevod onoga što naša muzika govori ili smo mi u funkciji ilustracije riječi zavisi šta je prije nastalo. Ako se radi muzika na tekst onda sam ja u funkciji priče koje postoji nastojim je ilustrirati sa adekvatnom melodijom ili harmonijama i tako a ako sam napravio melodiju ranije onda se tekstopisci obično jako muče jer je muzika šira. Jedan ton ima milion riječi koje ga mogu opisati a nijedna nije baš najpravija. Vrlo rijetko se dogodi da se napravi tekst na muziku pa da to bude savršeno poklapanje istovjetni ugođaj. Meni je draže raditi na tekst. Onda imam priču i znam točno koja je to boje jer je muzika isto boje.

Računari *Kako pevačice prihvataju računare? Postaju li i one hakerke?*

Rajko Dujmić (smeje se) Najteže nas prihvataju. Znate šta se događa sa pjevačima i pjevačicama? Oni dođu na kraju. Kada već to zvoni zvuči sve ima svoju atmosferu sve je jasno. Samo se fino uvale u jednu finu pannu debelu, onako (AAAAAH) ili se ne uvale! Onda smo mi vrlo žalosni. Jer je kasno da tražimo nove pjevače. Jer su pjesme napravljene za TOG pjevača. Mada nekih loših iskustava nemamo. S kim god sam radio uvijek je to namijenjeno tom čovjeku i nastojim se sa njim upoznati. Pjevač ili pjevačica, svejedno mi je. Mada više volim raditi sa pjevačicama.

Računari *Zašto?*

Rajko Dujmić Ženski glas mi je sugestivniji mogu više zahtijevati i raspon... Volim pisati za žene. Ljepše mi je pisati za žene. Osim toga ljepše mi je raditi u studiju sa njima. Ljepše ih je vidjeti meni (Dragan Čačinović se smeje i kaže da najviše voli raditi ploču za grupu „Stil". Zbog plesačica.) Žene su nekako organiziranije i emocionalno su drukčije. Mogu se prebaciti na razne emocije lakše nego muškarac i onda su ambiciozne sve. To ja vrlo poštujem. Strašno poštujem ambiciozne ljude. Jer šta će ti čovjek koji nije ambiciozan!

Računari *U kom smislu ambiciozan?*

Rajko Dujmić Da bude vrhunski u svemu što on radi. Znači na sebi ambiciozan. Ne ambiciozan na meni pa da mene tjera da je budem savršen pa da mi on ili ona to otpjeva sa lijevom nogom. Ne nego na sebi na svom izrazu na svom glasu da bude ambiciozan. To je vrlo bitno. Žene su vrlo ambiciozne i žele da to urade dobro i što najbolje mogu. Trude se i daju svoj optimum. A to je redovito dovoljno. Ja nemam puno iskustva sa muškarcima. Radio sem malo jedino što radim sa svojim dečkima iz grupe. Mi smo ultra profesionalna grupa i kompletno smo kao grupa ljudi koji žive od toga i dišu za to i ne damo se omesti. I sviramo za dvadeset kao i za dvije tisuće ljudi. Sasvim je svejedno koliko je ljudi i koja je dvorana i koja je zemlja.

• Pa, Dujmić ima najviše softvera u glavi, jer je znam šta hoću.

Računari *Ovaj studio je poseban. Savršena dvorana „Lisinski" sa kojom je povezan daje mu super resurse. Koliko se isplati ulaganje u kompjuterizovanu opremu koja preko noći postaje zastarela?*

Rajko Dujmić Filharmonija je nešto posebno. Danas je kompjuter najpovoljniji u pogledu cijene i softvera za ove primjene koja „nose" „stari".

Računari *Da li je skup softver?*

Rajko Dujmić Ne nije osnovni softver je trista maraka. To je područje na kojem stalno imate nešto novo. Računari se stalno razvijaju. Ja samo znam da čipovi iz „Space Shuttle" završe nakon pet godina u nekoj klavijaturi. Iskoriste ga za proizvodnju nekakvog zvuka.

Računari *Da sve se bitno promenilo od kad smo šetnju po mesečini zamenili šetnjom po Mesecu. Ima li pirata među muzičarima? Priča se da je to jedina korisnička grupa imuna na strasti piratisanja.*

Rajko Dujmić Da to je uglavnom tako mada sam čuo da ima slučajeva kopiranja.

Računari *Pitanje je i koliko se tako kopirani softver zaista koristi? Većina profesionalaca kupuje softver da bi imali najnovije verzije dokumentaciju.*

Rajko Dujmić Recimo „Štajnberga" se ne isplati kopirati jer to je softver vrlo profesionalan sa debelom pratećom knjižurinom.

Računari *To je onda i najbolja zaštita zar ne? Treba proučiti tu dokumentaciju!*

Rajko Dujmić Ipak kod nas ljudi koji imaju vrhunsku opremu i koji najbolje rade ne pirati. Ni jednog nisam video sa razbijenim „hard-lock-om"! Tu je uvijek bila i knjiga pokraj originala i originalni „key" i sve originalno.

OS/2

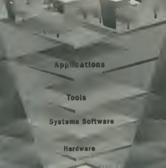
Usta puna hvale

Iako je OS/2 već neko vreme među nama, stiče se utisak da se ništa posebno od toliko najavljivanih revolucionarnih promena nije desilo. Utisak je zaista tačan — veoma mali broj mašina radi pod novim operativnim sistemom, a i broj važnih programa prilagođenih OS/2 može se na prste izbrojati. Pa ipak, iako ne tako nagle, promene će biti zaista duboke. Kreiran je sistem koji gotovo da izgleda kao i stari DOS, a ipak u sebi krije bezbroj novih mogućnosti na koje tek treba da se naviknemo.

Zoran Životić

Uobičajeno je u ovakim situacijama pred stavljanja novih proizvoda početi od hardverskih zahteva koji se pred sistem postavljaju. Kod OS/2 je ova tema pogotovo važna, ali ćemo ipak napraviti izuzetak i okrenuti redosled. Prvo ćemo pokušati da vam predstavimo šta je sve OS/2 u stanju da vam ponudi — tek na kraju ćemo načeti delikatnu temu — šta traži za uzvrat. U ovom prikazu ćemo se zadržati na verziji OS/2 1.0. Za verziju 1.1 i Presentation Manager će svakako biti dovoljno vremena i kasnije, pogotovo što se radi o nadgradnji — osnove koje nas ovde zanimaju nisu u novoj verziji ni malo promenjene. OS/2 nije glomazan operativni sistem.

Osnovne verzije zauzima svega 4 1,2 MB diskete, a sama instalacija je više nego jednostavna. U drajv se postavi instalaciona disketa i sistem resetuje. OS/2 se pokreće sa diskete i na hard disk prebacuje neku vrstu auto start instalacionog programa koji će preuzeti kontrolu kod ponovnog resetovanja računara i izvesti instalaciju do kraja. Već prvi korak će vas uveriti u staru tvrdnju da IBM pre svega vodi računa o kompatibilnosti sa ranijim proizvodima — instalacijom OS/2 svi podaci na hard disku ostaju netaknuti (ukoliko direktno ne zahtevate da se aktivne particija formatira), jedino se sistemske datoteke zamenjuju novim. Na hard disku će se naći nekoliko novih direktorijuma i zavidan broj novih datoteka u osnovnom direktorijumu. Posledica ovakve instalacije je da se OS/2 lako može ukloniti — dovoljno je da računar startujete sa DOS disketom i izvedete SYS C: i sve će se vratiti u poznato stanje. Kada nakon instalacije ponovo resetujete računar, nakon pozdravne poruke sa čuvenim IBM znakom na ekranu će se pojaviti takozvani „PROGRAM SELEKTOR" (slika 1). U levoj koloni je izbor programa koji se mogu startovati postavljanjem pokazivača i pritiskom na ENTER, dok se u desnoj koloni održava spisak programa koji trenutno rade. U desnoj koloni se uvek nalazi DOS Command Prompt kojim se aktivira poseban DOS kompatibilni režim.

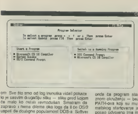


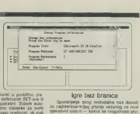
Ljubav na prvi pogled

Ako poželite da odmah isprobate pravi OS/2 režim, izabraćete iz leve kolone OS/2 Command Prompt i naići na tipičan problem — probati ali šta? Verovatno ćete početi sa komandom DIR i otkriti da OS ne zna ama baš ništa novo — spisak datoteka na ekranu je identičan onoma koji ispisuje DOS. I nekoliko drugih komandi (CHKDSK, CLS, COPY, XCOPY, TREE) vam neće otkriti nikakve novosti. Možda ćete osetiti mali trenutak radosti ako zadate formatiranje diskete — FORMAT A: /4 — a zatim pritisnete Ctrl + ESC. OS će vas prebaciti u program selektor gde možete ponovo u levoj koloni izabrati OS2 Command Prompt. Dobićete čist ekran, svež prompt i mogućnost da nastavite rad sa računarom — u pozadini će, ne ometajući vas, OS završiti započeti proces formatiranja diskete. S obzirom da se u njemu zahteva odgovor na pitanje o Volume Label u formatiranju još jedne diskete, možete se na ovaj proces vratiti na dva načina — prelaskom u selektor masku (Ctrl + ESC) i ukazivanjem na OS Command Prompt 1 u desnoj koloni ili korišćenjem kombinacije ALT + ESC koja će vas „prošetati" u krug po svim ekranima na kojima je započet odvojen posao. Koliko god ovakav primer bio atraktivan, teško da može predstavljati jak razlog za prelazak na novi sistem. U ovom trenutku vredi proveriti kako izgleda DOS režim — ako je kompatibilnost sa starim DOS-om potpuna, OS/2 bi mogao na neki način da bude u pozadini i da se postepeno aktivira kako budu stizale nove aplikacije pisane posebno za njega.

Prvi testovi će pokazati odlične rezultate — čak i delikatniji programi — razne rezidente rutine i direktni „stealci" na video memoriju rade bez problema. Ipak, sve ima svoje granice — programi koji smatraju da je čitav hardver isključivo na njihovom raspolaganju neće uopšte moći da rade. U mojoj praksi se priča svodi na samo jedan (ali vredan) program ovog tipa — *Ventura*. Dakle, razlog dovoljno jak da se ipak zamislimo o opravdanosti instaliranja OS u ovom trenutku. Odlučili smo da OS u ipak pružimo malo više šanse. Iako nema dvostruku BOOT opciju — alternativno DOS ili OS/2 — nije posebno teško napraviti DOS sistemsku disketu koja će u svemu vratiti računar u pravo DOS stanje, pogotovo što sve sistemske datoteke i dalje mogu da ostanu na hard disku.

Dakle, ostavljajući još uvek nezamenjiv DOS u rezervi, upustili smo se u malo dublju analizu onoga šta se stvarno dešava pod OS-





om Sve što smo od tog trenutka videli pokazalo je sasvim drugačiju sliku — sliku pred kojom će malo ko ostati ravnodušan. Smatram da zapravo i nema dileme oko toga da li će OS/2 uspeti da dostigne popularnost DOS-a. Softver na koji smo navikli je toliko dugo sa nama da nije ni malo čudno što niko ne želi da ga se odrekne. Prosto prevođenje na OS/2 je gotovo trivijalan posao, ali je činjenica da to ipak još uvek nije masovnije učinjeno. Čini mi se da razlog leži u činjenici da je zapravo šteta direktno prevesti aplikaciju samo zato da bi radila pod OS-om — programeru se pruže toliko novih mogućnosti da je za navikavanje na njih potrebno dosta vremena malo „drugačijeg" razmišljanja. Zato smo ubeđeni da je OS sigurno budućnost i da će aplikacije po mnogo čemu slediti samu OS filozofiju — spolja na izgled isto (u svakom slučaju poznato bez potrebe za novim znanjima), iznutra daleko fleksibilnije sa mnogo novih mogućnosti od kojih će isključivo korist imati krajnji korisnici.

Sistem

Osnovu OS/2, kao i DOS-a čine dve skrivene datoteke IBMBIO i IBMDOS, ali je sada ova druga daleko veća (oko 250 K). Inicijalizacija do određene faze je takođe identična, ali umesto konačnog startovanja COMMAND. COM interpretera sistem prelazi u pomenutu „Program selektor" masku. Za korisnika uočljiva razlika u inicijalizaciji je sada daleko složeniji CONFIG. SYS u kome se definišu svi globalni parametri sistema. Nećemo se ovde upuštati u objašnjenje pojedinih parametara — dovoljno je reći da i dalje važi stari sistem za instalisanje drajvera za pojedina uređaja putem DEVICE direktiva, da se i dalje može zadati alternativni SHELL (za DOS režim) i PROTSHELL za OS i da su novi parametri za raspodelu vremena (MAXWAIT, TIMESLICE itd) kao i nekoliko direktiva za kontrolu memorije i zadavanja staza do sistemskih datoteka. Vrednosti inicijalno postavljene instalacijom gotovo da nema potrebe menjati, a i kada se to čini, sva promena su veoma logična i ni malo komplikovane. S obzirom da se više komandnih procesora može startovati istovremeno, AUTOEXEC. BAT sada ima nešto drugačiju ulogu (i naziv — OS2INIT.CMD. CMD je inače nova ekstenzija za BAT datoteke) jer se automatski startuje tek kada iz selektor maske pokrenete OS2 Command Prompt, pa se zapravo odnosi lokalno na jednu seansu. S obzirom da se u njoj (kao i kod DOS-a) definiše PATH i niz drugih parametara koji su za različite zadatke različiti, OS2INIT nema tako veliki značaj. Naime delu je gotovo nemoguće „uglaviti" u nju PATH i SET-ove koji bi zadovoljili sve zahteve. Sada se kreira više različitih datoteka ovog tipa, a poziv svake od njih se obezbeđuje iz selektor maske. Konkretno, ako jedna od seansi treba da bude rad sa kompajlerom, treba kreirati recimo CINIT.CMD sa SET, PATH naredbama i slično i zatim dodati poziv ovog programa u selektor masci (vidi sliku 2).

Umesto CMD datoteke iz selektor maske se može direktno startovati i program, ali se ovo pokazalo prilično nepraktično. Ako na primer ugradite direktan poziv editora MEP.EXE onda će program startovati u praktično praznom okruženju — bez definisanih SET-ova ili PATH-ova koji su mu potrebni. Sistem automatskog startovanja jedne datoteke za svaki posao odvojeno ima mnogo prednosti, ali ipak nedostaje bar jedna globalnog tipa. Recimo, treba pozvati program za postavljanje brzine auto—ponavljanja tastatura koji očigledno ima smisla samo za čitav sistem i treba biti izveden odmah po inicijalizaciji računara. Ovaj nedostatak se idealno rešava RUN direktivom u CONFIG. SYS datoteci, na primer:

RUN=C:/OS2/KEYRATE.EXE 21 250

ali je za sada po malo nejasno šta program startovan na ovaj način sme da radi (s obzirom da sistem u tom trenutku nije do kraja završio inicijalizaciju). Na ovaj način je rešeno aktiviranje na primer PRINT SPOOL programa.

Postoje četiri načina da se pod OS-om startuje program. Jedan smo opisali — dodavanja naziva i startovanje iz selektor maske — dok se sledeća tri izvode direktno iz komandnog prompta. Pored „normalnog" navođenja naziva programa na raspolaganju nam je i naredba START:

START MEP

Ovo će startovati program MEP, dok će se na ekranu odmah ponovo pojaviti prompt, mogućnost za nastavak tekućeg posla. Prelazak u ovako startovan program se izvodi bilo putem program selektora (njegov naziv se pojavljuje u spisku aktivnih programa) bilo „kruženjem" po seansama kombinacijom Alt+ESC. Sledeći način je po svemu specifičan i izvodi se naredbom DETACH, na primer:

DETACH XCOPY D:\DLL*.* A:\

Nakon ovoga, prompt i kursor će se odmah pojaviti i može se nastaviti tekući posao. Ako pokušamo da aktiviramo ekran posvećen ovom poslu ni jednom metodom nećemo uspeti da ga nađemo! DETACH je zapravo način za startovanje programa koji ne zahtevaju nikakvu interakciju sa korisnikom, odnosno ekran i tastature im nisu na raspolaganju. Da smo na isti način startovali naredbu FORMAT A: ne bismo uspeli da je izvedemo do kraja jer bi ona uzalud očekivala da pritisnemo RET nakon poruke „Insert new disk in drive ..." Ipak i naredba FORMAT se može startovati na ovaj način — dovoljno je da kreiramo datoteku na primer FMT.ANS u kojoj su redom zapisani odgovori na pitanja koje program postavlja i izvedemo:

DETACH FORMAT A: <FMT.ANS >NUL

Kao što smo već spomenuli, OS2 raspoznaje gotovo sve naredbe na koje smo navikli kod DOS-a i izvodi ih na identičan način. Ipak, ugrađeno je mnogo novih nijansi koje nisu vidljive na prvi pogled. Na primer, standardno je proširen broj argumenata koje pojedina naredba primaju, pa je sada moguće:

DEL C:*.BAK D:*.BAK D:\TEMP*.TMP ili DIR *.EXE *.CMD

Otvorenoj mogućnosti za složenije komanda ni malo ne odgovara način editovanja komandnih linija koji je (što gotovo da nismo mogli da poverujemo) ostao kao i kod DOS-a sa čuvanim F1, F2 i F3 tasterima!

Igre bez granica

Spominjanja ovog nedostatka nas dovodi do najinteresantnijeg pitanja vezanog za novi operativni sistem — kakva se mogućnosti pružaju programerima u oblastima (1) uobičajenog programskog korišćenja resursa računara i (2) prilagođavanja i korekcije operativnog sistema putem raznih sistemskih programa na koje smo svi bez izuzetka navikli.

Odgovor na ova pitanja je — izvanredna i gotovo neograničena! Kao primer navešću podatak da je, uz nešto obavezne literature, programeru dovoljno jedno popodne da napravi osnovnu verziju (i da je pokrenali) programa tipa CED (editora komandne linije koji dozvolja ponavljanje prethodno izvedenih naredbi, normalno editovanje tekuće linije, upotrebu sinonima itd). OS2 najveći deo svog sistema izvodi iz takozvanih DYNAMIC LINK datoteka (DLL — zbirka rutina koje se pozivaju na zahtev programa tek u fazi njegovog izvođenja). Svaka funkcija iz ovih biblioteka može biti zamenjena sopstvenom rutinom koja se takođe kreira u DLL formatu. S obzirom da sistem za editovanje komandne linije koristi funkciju KbdStringIn iz KBDCALLS.DLL biblioteka, više je nego jednostavan zadatak izvršiti zamenu i instalaciju CED-a. Ono što posebno raduje je da broj ograničenja koja sistem nameće u smislu šta ovakav program sme da radi (glavna tema kod DOS sistemskih rutina) nije velik, pa programi mogu da budu veoma fleksibilni.

Jedan od veoma popularnih tipova programa takozvana rezidentne rutina (SideKick kao najpoznatiji) koje su autonomne pod DOS-om zadavala velike glavobolje (a verovatno se i DOS slično „osećao" dok ih izvodio), sada na neki način dobijaju pun legitimitet. Implementacija se ostvaruje putem takozvanih „Device Monitors" — programa koji se vezuju u ulančanu listu i koji redom primaju, analiziraju i preuzimaju ili prosleđuju pakete informacije od pojedinih uređaja. Ono što se PCTOOLS sa svojim DESKTOP-om trudi da postigne (pa i mnogi drugi) sa vrlo sumnjivim rezultatom pod DOS-om, pod OS-om postaje veoma jednostavno. Ne možemo se oteti utisku da će vrlo brzo rad sa OS2 postati pravi praznik pravog testera — sve što nam je potrebno biće zaista na dohvat ruke.

Ako je OS2 „raj" za programe sistemskog tipa, nije ništa manje komforan ni za „obične" programa. Jedna od glavnih prednosti je mnogo veća količina informacija koju je sistem u stanju da pruži kao i konačno razbijanje barijere „640K". Iako je format zapisa datoteka i direktorija ostao potpuno kompatibilan sa DOS-om, OS2 će vam pružiti mnogo više informacija — tako se pored datuma i vremena kreiranja datoteke može dobiti informacija o datumu i vremenu poslednjeg pristupa podacima, datumu i vremenu poslednjeg zapisivanja u datoteku, pa čak i prostoru koji datoteka na disku stvarno zauzima (s obzirom na klaster kao alokacionu jedinicu ova je vrednost uvek veća od stvarne dužina datoteke). Sa tastaturom je situacija još lepša — u jednom pozivu popunjava se struktura iz koje se može saznati ASCII kod, sken kod, stanje shift tastera da li je ta-

Nastavak na str. 21

PASCAL ILI MODULA

Modula 2 je programski jezik o kome se tokom proteklih godina dosta govorilo i pisalo ali u kome se, po svoj prilici, nije baš previše programiralo. Treba li, dakle, savetovati prosečnom PC programeru da pređe sa *Turbo Pascal*-a na Modulu 2?

Dejan Ristanović

Iako poređenje performansi pojedinih programskih jezika nije ni malo jednostavno, takmičenju paskala i Module 2 nema previše mesta — oba jezika je izmislio isti čovek pri čemu je Modula 2 nastala docnije kao usavršeni paskal, to bi *a priori* trebalo da znači da je ona bolja! Cilj ove naše analize, međutim, nije da diskutujemo o tome da li je Modula 2 kao jezik bolja od paskala ili nije. Interesuje nas kakve dobitke i gubitke može da očekuje jedan **prosečan PC programer** kada odluči da napusti *Turbo* Pascal i pređe na *Logitech* ili *JPI* Modulu 2 — ne govorimo, dakle, o jezicima nego o njihovim konkretnim implementacijama!

Diskusija o dobicima i žrtvama prosečnog PC programera mora da započne nekom predstavom o njegovim mogućnostima i potrebama. Prosečnog PC programera zamišljamo kao čoveka koji piše manje, srednje ili (ponekad) velike programske pakete koji treba da obave neki konkretan posao, koji radi samostalno ili u timu sa još jednim kolegom, koji ime standardnu XT ili AT kompatibilnu konfiguraciju i koji preferira korišćenje funkcija jezika nad komunikacijom sa DOS-om ili BIOS-om. Naš prosečni programer se takođe ne zanosi multiprogramskim radom, pisanjem rezidentnih programa ili direktnom komunikacijom sa disk kontrolerom — cilj mu je prvenstveno da što pre uredi posao koji je započeo!

Za ovako definisanog prosečnog programera karakteristike samog jezika su važna ali ne uvek i presudna stavka — od velikog je interesa komfor u radu jer ne može biti svejedno da li će proces prevođenja i povezivanja programa trajati pet sekundi ili pola minuta, od značaja je i kvalitet debagera, broj biblioteka sa potprogramima, dokumentacija i mnogi slični parametri koji su direktno vezani za kvalitet određene implementacije paskala odnosno Module 2. Zato smo odlučili da umesto paralela opšteg tipa poredimo konkretne implementacije. Što se paskala tiče, u konkurenciji je jedino Borlandov *Turbo Pascal 5.0* dok ćemo što se Module 2 tiče koristiti dve najkvalitetnije varijante — *Logitech Modula 2 3.0 i JPI (Top Speed) Modula 2 1.12*. Neposredne karakteristike ovih programa ćemo, međutim, upoznati za mesec dana — ovoga puta moramo da obratimo pažnju na razlike između paskala i Module 2 kao jezika.

Velika i mala slova

Prva razlika koju ćete primetiti kada pogledate program pisan na paskalu i program pisan na Moduli 2 je pravopis — dok u paskalu (samim tim i u Turbo Pascalu) možete slobodno mešati velika i mala slova i po želji pisati Begin, BEGIN, begin ili čak bEgin, u Moduli 2 se zna da sve ključne reči moraju da se pišu velikim slovima — BEGIN i tačka. Uobičajeno je, uz to, da se imena promenljivih pišu malim slovima što bi moglo da olakša snalaženje u programu. Kod Module 2 su, međutim, razlike između velikih i malih slova dosledno poštovane: ne samo što se mora pisati BEGIN i što se promenljive Prvi i PRVI razlikuju, nego se i imena bibliotečkih funkcija moraju pisati tačno onako kako su definisana — nije dosta što pamtimo da se funkcija za čitanje racionalnih brojeva zove *rdreal (JPI Modula 2)* nego moramo i da se setimo da je ime baš *RdReal* a ne, na primer, *Rdreal* ili *RDREAL*. Čovek se na kraju navikne na ovaj teror malih i velikih slova ali mu u početku baš i nije lako!

Nesumnjiva prednost koju ćete osetiti čim pređete na Modulu 2 je slobodan prostor — dok je na *Turbo Pascal*-u segment podataka ograničen na 64 kilobajta, u Moduli 2 možete da se „prostrete" preko čitave raspoložive memorije. Turbo Pascal, naravno, omogućava korišćenje slobodne memorije za smeštanje dodatnih podataka (možemo se, na primer, snaći kao na slici 1) ali je svaka struktura ograničena na 64 K — razni trikovi zahtevaju određeno procesorsko vreme što nužno usporava programe sa velikim memorijskim zahtevima. Na Moduli 2 jednostavno možete da date „pun gas".

Sledeći dobitak su stringovi čija je dužina ograničena na 65535 a ne 255 znakova. Programeri — početnici često smatraju da je *Turbo Pascal*-ovih 255 znakova sasvim dovoljno za svaku primenu ali ovakvo rezonovanje pada u vodu čim na bilo koji način počnemo da obrađujemo tekst — jedinica obrade je obično jedan pasus koji prirodno može da bude relativno dugačak: pasusi od 5—6 kilobajta ne predstavljaju posebnu retkost. Paskal je naravno dobro struktuiran jezik na kome nije teško realizovati procedure koje rade sa stringovima dužim od 255 znakova, ali je mnogo pogodnije kada su takve procedure ugrađene u osnove jezika.

Procedure i funkcije

Kada bismo pokušali da realizujemo procedure za rad sa drugim stringovima na paskalu, vrlo brzo bismo se suderili sa jednom zanimljivom preprekom. Dugom stringu smo, na primer, dodelili tip *lstring* deklarisan kao *array [1..maxstr] of char*. Zaključili smo, osim toga, da nam je potrebna funkcija lcopy (ekvivalent *Turbo Pascal* funkcije *copy*) i napisali nešto poput

function lcopy (s: lstring; odakle, koliko: word): lstring

Kompajler se, međutim, pobunio — funkcija u paskalu može da „vrati" samo celu, racionalnu, Bulovu ili znakovnu vrednost odn. tipovi koje smo definisali mogu da karakterišu argumente ali ne i rezultate! Ovo je velika nedoslednost paskala (i *Turbo Pascal*-a) sa kojom se *Wirth* verovatno morao pomiriti u cilju pojednostavljenja kompajlera. Dana u kojima je nastajala Modula 2 karakterisali su računari većih mogućnosti pa je nedostatak uklonjen — nema nikakvog problema da programirajući na Moduli 2 u glavni program vratite rezultat proizvoljnog tipa. Slika 2 demonstrira ovu tehniku — program koji sabira kompleksne brojeve deklarisane kao slogove koji se sastoje od realnog i imaginarnog dela, pri čemu je funkcija *cadd* definisana kao

PROCEDURE cadd (a,b: complex): complex

Rekli smo funkcija a napisali PROCEDURE — zar to nije neobično? Modula 2 jednostavno ne obuhvata službenu reč FUNCTION već se procedure dele na one koje (sa *RETURN* vrednost) nešto vraćaju i one koje samo modifikuju argumente — razlika je isključivo kozmetičke prirode. Postoje, međutim, i mnoge druge razlike između paskalove i Modulne koncepcije rada sa procedurama čiji je značaj daleko veći.

Osnovna koncepcija Module 2 zahteva da se problem koji rešavamo podeli na logičke celine i da se onda svaka od tih celina realizuje posredstvom jednog modula: moduli se docnije posebno testiraju da bi se centralna aplikacija na kraju sastojala samo od nekoliko poziva glavnih potprograma. Rad sa posebnim modulima koji su upisani u posebne datoteke donosi još jednu indirektnu korist — pošto popravimo jedan potprogram, moramo da prevodimo samo njega i njemu nadređene module čime se znatno štedi na vremenu: potrebno je jedino obezbediti da linkovanje (tj. povezivanje modula u finalni EXE program) ne traje predugo.

Razbijanje aplikacije u posebne datoteke je velika prednost Module 2 nad paskalom ali ne i nad *Turbo Pascal*-om — počevši od verzije 4.0 *Turbo Pascal* podržava module koji se (automatski) nezavisno prevode i upisuju u posebne TPU datoteke koje (od korisnika dobro skrivene) linker onda povezuje gotovo neverovatnom brzinom. *Turbo* Pascal je preuzeo još po nešto od Module 2: svaki modul može da ima svoje javne konstante, promenljive, procedure i funkcije (deklarisane u segmentu INTERFACE) kao i privatne konstante, promenljive, procedure i funkcije (segment IMPLEMENTATION) koje su nadređenim modulima nedostupne. Modula 2, međutim, krije još bar dve prednosti koje ćemo ponovo objasniti na primeru programa koji radi sa kompleksnim brojevima.

Modul *ComplexOps* sa slike 3 sadrži procedure za sabiranje i oduzimanje kompleksnih brojeva pri čemu je tip *complex* definisan u okviru definicione sekcije tj. pristupačan svim nadređenim modulima — glavni program može da sabere dva kompleksna broja kako pozivom funkcije *cadd* tako i prostim

c3.real := c1.real + c2.real
c3.imag := c1.imag + c2.imag

Program sa slike 4 je malo drugačiji — u definicionoj sekciji je tip *complex* samo pomenut dok je njegov precizan opis dat u radnoj sekciji — glavni program i dalje može da deklariše promenljive tipa *complex* i da obavlja njihovo sabiranje pozivajući cadd ali je tačna struktura promenljive nepoznata što znači da bi naredbe poput

c1.real := 3.5

rezultirale greškom. Programer, dakle, može da koristi promenljive samo na način koji je autor modula predvideo!

Kakva je korist od ovakvog ograničenja? Zamislimo projekat na kome radi veći broj ljudi od kojih je jedan zadužen za operacije sa kompleksnim brojevima. U početku je napisan modul poput onoga sa slike 3, pisani su i moduli koji ga pozivaju i najzad je zaključeno da je pogodnije da se kompleksni brojevi zadaju u polarnim koordinatama. Modul *ComplexOps* je na odgovarajući način prepravljen i neki rađe dani moduli su odmah prestali da funkcionišu — sabiranje polarno zadatih kompleksnih brojeva se ne može svesti na prosto sabiranje realnih i imaginarnih delova! Ukoliko je međutim konačno rešenje sa slike 4 bilo kakve promena u potprogramu neće izazvati posest među nadređenim jedinicama — kompleksnim brojevima smo i onako mogli da pristupamo samo preko procedura što znači da promenu u nekoj od njih nemamo mnogo šansa da primetimo.

Nesreća je jedino u tome što (i dalje) radimo sa računarima ograničene memorije uvek sumnjive brzine — svaki poziv procedure ili funkcije zahteva neko vreme što znači da će direktan pristup poljima kompleksne promenljive ubrzati rad programa. Ukoliko dakle sami redimo na Moduli 2 njene pikanterije ovoge ti-pa verovatno nećemo ni koristiti — mehanizmi *Turbo Pascala* su sasvim dovoljni.

Prenos argumenata nije ni druga vidljiva razlika između *Turbo Pascal* i Module 2 — glavni program sa slike 3 je „uvezao" samo proceduru *cadd* što rezultuje kraćim EXE programom. Slično se manje-više ponaša i Borlandov linker ugrađen u *Turbo Pascal* s tim što je čitav red transparentan za korisnika — dok na Modul 2 moramo stalno da pišemo šta se odakle „uvozi" Turbo Pascal će automatski povezati samo ono što je potrebno!

Sa slike 3 i 4 vidi se još jedna razlika između *Turbo Pascal*-a i Module 2: INTERFACE i IMPLEMENTATION segmenti (da ostanemo kod *Turbo Pascal* terminologije, u Moduli 2 se INTERFACE segment zove DEFINITION MODULE) su potpuno odvojeni i svaki se smešta u posebnu datoteku koja se posebno prevodi. Zašto ova komplikacija? Ideja se opet temelji na interesima veće grupe programera koji rade na jednom projektu — u početku se dogovori koji će moduli postojati, koje će tipove procedure i funkcije koji od njih „izvoziti" i kako će se redati argumenti pojedinih potprograma. Onda se napišu definicioni moduli i izrada svake implementacije prepusti nekom od programera — bilo koja izmena u implementacionom modulu se praktično ne održava na nadređene module koji „vide" jedino DEFINITION. samo kada se zaključi da početnom dogovoru nedostaje neka procedura ili neki argument u nekom od modula (previše ovakvih situacija znači da je posao loše isplaniran) ukazaće se potreba za ponovnim prevođenjem čitavog programskog sistema.

Iako u osnovi logična, ova koncepcija predstavlja pomalo otežavajući faktor za našeg prosečnog programera — treba misliti o tome da li su prevedena dva modula, da li imena odgovaraju, da li su tekstovi u njima usaglašeni i tome slično. Kod *Turbo Pascal*-a je kompletan kod koji pripada modulu u jednoj datoteci pri čemu je linker veoma pametan — ako promenite implementacioni deo procedure ALFA iz modula KONTROLA, biće ponovo prevedeni samo oni moduli koji koriste ALFA a ne i oni koji koriste neki drugi potprogram iz modula KONTROLA — kada se ova karakteristika „saberе" sa izvanrednom brzinom prevođenja koju *Turbo Pascal* nudi (više o njoj sledećeg meseca) verovatno će se pokazati da razdvajanje implementacione i definicione sekcije donosi više glavobolje nego koristi.

```
Slika 1

program heap;

{
    Preko 64 K u Turbo Pascalu
            - demo -
}

uses crt;

const maxstr = 60000;

type dugistr   = record
                   duzina: longint;
                   txt: array [1..maxstr] of char
                 end;
     ukazt     = ^dugistr;

var ukaz: ukazt;
    c     char;

procedure dodeli (ukaz: ukazt; koliko: longint);
var i  longint;
begin
   ukaz^.duzina:=0;
   for i:=1 to koliko do
   begin
      ukaz^.txt[i]:=chr (random (96)+32);
      inc(ukaz^.duzina);
   end;
end;

procedure ispis (ukaz: ukazt);
var i: longint;
begin
   for i:=1 to ukaz^.duzina do
       write (ukaz^.txt[i]);
   writeln;
end;

begin
   getmem (ukaz, sizeof (dugistr));
   dodeli (ukaz, 50000);
   ispis  (ukaz);
   repeat until keypressed;
   c:=readkey
end
```

```
DEFINITION MODULE ComplOp;                     Slika 3

   TYPE complex = RECORD
                     realni: REAL;
                     imagin: REAL;
                  END;

   PROCEDURE cadd  (a, b: complex): complex;
   PROCEDURE cubtr (a, b: complex): complex;

END ComplOp.

IMPLEMENTATION MODULE ComplOp;

   PROCEDURE cadd (a, b: complex): complex;
   VAR rez: complex;
   BEGIN
      rez.realni:=a.realni+b.realni;
      rez.imagin:=a.imagin+b.imagin;
      RETURN rez;
   END cadd;

   PROCEDURE cubtr (a, b: complex): complex;
   VAR rez: complex;
   BEGIN
      rez.realni:=a.realni+b.realni;
      rez.imagin:=a.imagin+b.imagin;
      RETURN rez;
   END cubtr;

END ComplOp.

MODULE glavni;
FROM IO IMPORT RdReal, WrReal;
FROM ComplOp IMPORT complex, cadd;

VAR c1, c2, c3  complex;

BEGIN
   c1.realni:=RdReal ();
   c1.imagin:=RdReal ();
   c2.realni:=RdReal ();
   c2.imagin:=RdReal ();
   c3:=cadd (c1, c2);
   WrReal (c3.realni, 10, 3);
   WrReal (c3.imagin, 10, 3);
END glavni.
```

Ulazno-izlazni problemi

Veliki argument u prilog Modula 2 je postojanje brojnih komercijalno raspoloživih modula koje možete po potrebi uključivati u aplikacije koje pišete — dosta ovakvih potprograma dobijate uz sam kompajler, naročito ako ste se odlučili za *Logitech*-ovu Modulu 2. Moramo, međutim, da kažemo da će vas prvi dani rada sa modulima isterati iz kože — ne samo što na početku programa morate da definišete sve promenljive koje ćete koristiti nego morate da navedete i listu procedura ne zaboravljajući iz kog modula se koja procedura uvozi. Kada bi se na ovaj način „uvozile" samo egzotičnije procedure, sve bi još i bilo prihvatljivo, nevolja je, međutim, što elementi Module 2 nisu čak ni najelementarnije funkcije za ulaz i izlaz brojeva i stringova — da bi ste izvršili obično *writeln (Kraj rada)* morate da uvezete proceduru *WrStr* iz modula IO. Ako vam je uz to potrebno da štampate i neki ceo broj, trebaće vam procedura *WrInt* a možda i *WrLngInt*, za racionalne brojeve se koriste procedure *WrReal* i *WrLongReal* (imena važe za JPI Modulu 2, kod *Logitech*-ova se na primer koristi *WriteReal* čiji su argumenti isti — toliko što se tiče

```
Slika 2:

TYPE complex = RECORD
                  realni. REAL,
                  imagin  REAL,
               END,

PROCEDURE cadd  (e, b. complex): complex,
VAR rez: complex,
BEGIN
   rez realni =a.realni+b realni,
   rez imagin =a imagin+b.imagin,
   RETURN rez;
END cadd,
```

```
slika 5:

MODULE kontrol,

FROM IO IMPORT WrCard, RdCard, WrLn, WrStr,

VAR i: CARDINAL;

BEGIN
   FOR i =1 TO 10 BY 2 DO
       WrCard (i,5);
       WrLn,
   END;
   WrLn;
   i:=1;
   LOOP
      WrCard (i, 5),
      WrLn,
      IF    i=7 THEN
            EXIT;
      ELSIF i=4 THEN
            WrStr ( --- )
            WrLn
      END,
      INC (i),
   END;
   REPEAT
      WrLn;
      WrStr ( Unesi broj izmedju 1 i 3 (0 za kraj): )
      i =RdCard ();
      CASE i OF
           0:
           1:   WrStr ( Broj jedan );
           2:   WrStr ( Broj dva );
           3:   WrStr ( Broj tri )
           ELSE WrStr ( Neispravan unos ),
      END
   UNTIL i=0
END kontrol
```

```
Slika 4:

DEFINITION MODULE ComplOp;
   TYPE complex;

   PROCEDURE cadd  (a, b: complex): complex,
   PROCEDURE cubtr (a, b: complex): complex,
   PROCEDURE cread ():              complex;
   PROCEDURE cwrite (cc:  complex);
END ComplOp


IMPLEMENTATION MODULE ComplOp;
   FROM IO IMPORT RdReal, WrReal, WrStr, WrLn,
   FROM Storage IMPORT ALLOCATE,

   TYPE complex = POINTER TO RECORD
                     realni: REAL;
                     imagin: REAL,
                  END,

   PROCEDURE cadd  (a, b. complex): complex,
   VAR rez· complex;
   BEGIN
      NEW (rez);
      rez^.realni:=a^.realni+b^ realni;
      rez^ imagin:=a^ imagin+b^.imagin;
      RETURN rez;
   END cadd,

   PROCEDURE cubtr (a, b: complex): complex;
   VAR rez: complex,
   BEGIN
      NEW (rez);
      rez^ realni:=a^ realni+b^.realni,
      rez^.imagin:=a^.imagin+b^ imagin,
      RETURN rez,
   END cubtr,

   PROCEDURE cread(). complex;
   VAR rez. complex;
   BEGIN
      NEW (rez);
      WrStr ('Realni deo  ');
      rez^ realni.=RdReal ();
      WrStr ('Imaginarni deo  ');
      rez^ imagin:=RdReal ();
      RETURN rez,
   END cread,

   PROCEDURE cwrite (cc  complex);
   BEGIN
      WrReal (cc^.realni, 4, 10);
      WrStr  (  +  ),
      WrReal (cc^ imagin, 4, 10);
      WrStr  ('*j )
      WrLn,
   END cwrite;
END ComplOp.


MODULE glavni;
FROM ComplOp IMPORT complex, cadd, cread, cwrite
VAR  c1, c2, c3  complex;
BEGIN
   c1 =cread();
   c2.=cread(),
   c3 =cadd (c1, c2);
   cwrite (c3);
END glavni.
```

portabilnosti Module 2) i tako dalje — treba vam ukratko čitav modul IO. Ukoliko međutim program započnete sa IMPORT IO a ne sa FROM IO IMPORT WrStr, WrInt ... docnije WrStr ćete morati da zamenite sa IO.WrStr čak i u slučaju da procedura WrStr ne postoji ni u kom od preostalih uvezenih modula!

Potreba za navođenjem lista procedura koje se uvoze je mala neprijatnost u poređenju sa činjenicom da nemate naredbe read i write — umesto da napišete writeln ('Obrađeno', n, 'zapisa') moraćete najpre da pozovete proceduru WrStr i ispišete tekst Obrađeno, zatim proceduru WrInt i ispišete broj n i najzad novo WrStr koja će ispisati reč zapisa — tri poziva umesto jednog! Modula 2 je, kada se radi o ulazu sa tastature i izlazu na ekran, daleko nekomfornija od svih drugih popularnih jezika (čak i od jezika mnogo nižeg nivoa kao što je C) — želja da se bude potpuno nezavisno od hardvera (promenite modul IO i svi programi će bez problema „ispisivati" na neku novu jedinicu) rezultirala je potrebom da se sasvim jednostavne operacije dele na nekoliko (čak desetak) programskih redova čime se na kraju krajeva smanjuje preglednost programa i otežava njegovo debagovanje.

Kontrolne strukture

Modula 2 je, kada se radi o kontrolnim strukturama, bogatija od Turbo Pascal-a što ilustruje i slika 5: for petlji je dodat STEP dobro poznat bejzik programerima, uvedena je „mrtva petlja" LOOP iz koje se izlazi sa EXIT, IF ima klauzulu ELSIF (else if) a case ima else (ovo poslednje je unapređenje u odnosu na standardni paskal ali ne i na Turbo Pascal).

Najzanimljivija je međutim ukidanje stalnih potreba za službenom reči BEGIN — podrazumeva se da će se iza DO ili THEN naći niz instrukcija koje čine telo petlje i koje se završavaju sa END: početno BEGIN je odbačeno kao nepotreban balast što povećava komfor pri radu. Negativna posledica je što se i IF čija ELSE klauzula ima samo jednu naredbu mora završiti sa END ali nam se čini da je dobitak vredniji od žrtve.

Jedina prednost Turbo Pascala u odnosu na Modulu 2 je naredba goto koja je možda odgovorna za nečitke bejzik programe pisane u „špageti stilu" ali čija kontrolisana upotreba može da olakša pisanje pojedinih aplikacija: ponekad je jednostavno goto preglednije od višestrukih ugnježdenih IF-ova! Nedostatak naredbe goto se može staviti na dušu Wirth-u i Logitech-u ali ne i firmi Jensen & Partners International — JPI Modula 2 podržava goto i label kao ekstenziju.

Aritmetički i logički izrazi pretrpeli su male ili nikakve izmene — prioritet operatora je takav da se mora pisati IF (a>b) AND (c>d) THEN ... a ne kao u većini drugih jezika IF a>d AND c>d THEN ... Zanimljivo je uz to da Modula 2 poput paskala i Turbo Pascala nema operator za stepenovanje: zaista nam nije jasno zašto je ova operacija proglašena toliko nepotrebnom!

Što se tipova podataka tiče, Turbo Pascal i Modula 2 su relativno slični jezici — i pored pretenzija na mašinsku nezavisnost, autori implementacija moraju da se orijentišu na konkretan procesor i njegov aritmetički koprocesor što znači da svi dobri PC jezici rade sa slično definisanim podacima. Obzirom da su tipovi podataka u mnogo čemu stvar implementacije jezika, bavićemo se njima kada budemo poredili Logitech Modulu 2 i Turbo Pascal.

Poslednja (kako se smatra velika) prednost Module 2 nad drugim jezicima je mogućnost paralelne obrade — zahvaljujući modulu Processes možemo da „lansiramo" nekoliko procesa koji će prepuštati kontrolu jedan drugome uz potrebnu sinhronizaciju i nadzor od glavnog programa. Ovo divno zvuči ali u stvari nije ništa posebno — kada se sve sabere i oduzme ispada da je jedini način da jedan proces prepusti kontrolu drugome izvršavanje neke procedure koja liči na GOTO — kada prvi program dođe do neke tačke, izvršiće GOTO drugi i čekati da drugi izvrši GOTO prvi! Multiprogramski rad je, što se već više puta pokazalo, stvar operativnog sistema a ne jezika što znači da je mnogo lakše pisati multiprogramske aplikacije na nekom arhaičkom VAX-ovom FORTRAN-u nego na najmodernijoj PC Moduli 2.

Kada se sve sabere i oduzme, jedina velika prednost Module 2 nad Turbo Pascal-om je pristup kompletnoj memoriji i rad sa dugim stringovima, velike prednosti Turbo Pascal-a je normalan (fortranovski) tretman velikih i malih slova i postojanje komfornih procedura za ulaz sa tastature i izlaz na ekran. Ostale karakteristike ova dva jezika se mnogo ne razlikuju što znači da konačan prelazak sa paskala na Modulu 2 može da bude motivisan jedino kvalitetom implementacije nekog od ovih jezika: zato ćemo u sledeća dva broja „Računara" uporediti Turbo Pascal sa Logitech-ovom i JPI Modulom 2.

Usta puna hvale

nastavek sa str. 17

stei pritisnut ili otpušten, kao i vremenska oznake za proračun razmaka od prethodnog pritiska na taster.

Jedan jednostavni eksperiment

Već duže vreme intrigirani činjenicom da naš računar raspolaže sa čitavih 4 megabajta memorije a da se sa njom ne može gotovo ništa uraditi, bili smo posebno zainteresovani da vidimo kako će teći alokacija velike količine memorije pod OS-om. Jedan mali C program nas je brzo uverio da je i ovde situacija odlična.

```
#include <malloc.h>
main ( )
{
    long space =0
    while (malloc(Oxfff0)!=NULL)
    printf( Alocirano %1d/n' space+ =Oxfff0)
}
```

Sistem je munjevito stigao do cifre od oko 2.5M! Zatim je nastalo naglo usporenje ali je cifra i dalje nastavila da se povećava — program je stao tek nakon 8MB. OS2 koristi takozvanu virtuelnu memorijsku šemu koja uz DISK SWAP mehanizam dozvoljava adresiranje 1 gigabajta, ali je u praksi ograničena raspoloživim disk prostorom (32M). Čitav mehanizam je apsolutno transparentan za program ali će ipak biti potrebe za malo pažljivijim baratanjem velikim količinama memorije. Iako se u našem računaru nalazi prilično brz hard disk (18ms), prenos memorijskih segmenata na disk i sa njega je ipak spor proces. Nepažljivim pisanjem programa može se desiti da na primer pristup jednom elementu niza zahteva prebacivanje čitavog segmenta sa diska. U takvim situacijama sistem je strahovito spor, čak toliko spor da se ne isplati sačekati ni jedan prolaz programa, već ga odmah treba izmeniti!

Opšti zaključak

Iako bismo mogli još dugo da nabrajamo lepe strane OS-a (a neke stvarno važne nismo ni spomenuli), ipak ćemo zastati i pokušati da izvedemo opšti zaključak.

OS2 je bez sumnje odličan operativni sistem. Ono što posebno impresionira je činjenica da se pod starim DOS izgledom krije nešto potpuno drugo. Znanje stečeno pod DOS-om je velikim delom primenjivo na OS i na osnovnom korisničkom nivou prelazak je potpuno bezbolan. Sa programerske tačke gledišta otvara se mnogo novih mogućnosti — sa dobrim kompajlerom kao što je Microsoft C5.10 i nešto dokumentacije (svakako preporučujemo Norton Guides Kernal API kao veoma dobar iako nedovoljan) vrlo se brzo prelazi na novi sistem.

S obzirom da, bar kod nas, još uvek nema mnogo pravih OS aplikacija, moglo bi se reći da u ovom trenutku ipak mali broj korisnika može imati stvarne koristi od novog sistema. Tu pre svega mislim na programere koji dobijaju izvanredno okruženje za razvoj programa — ne samo profesionalnih namenjenih tek otvorenom OS tržištu, već i onih za internu upotrebu — razvijanje i rad sa programima za razne inženjerske proračune u novom okruženju predstavlja pravo zadovoljstvo. Nedostatak nekih programa može se ipak dovoljno efikasno nadoknaditi DOS kompatibilnim modom — dok kompajler izvodi svoj posao, ništa vas ne sprečava da se vratite u svoj omiljeni DOS editor i nastavite razvoj programa.

Strogo sa hardverom

I na kraju, veoma važno pitanje. Poznato je da OS/2 zahteva AT kompatibilni računar zasnovan na procesoru 80286 ili 80386. Iako se pod kompatibilnošću sa IBM AT do sada moglo podrazumevati mnogo stvari (kompatibilnost na nivou DOS-a, BIOS-a itd) od ovog trenutka će samo jedan podatak biti presudan — POTPUNA HARDVERSKA kompatibilnost. Razlog je prilično jednostavan. S obzirom da OS/2 radi u zaštićenom režimu rada 80286 originalni BIOS mu nije na raspolaganju pa najniži nivo operativnog sistema direktno upravlja hardverom. Dizajn osnovnih ploča uglavnom ne predstavlja problem jer su zaista kompatibilne sa svojim uzorom (mada ovakvu tvrdnju ne bi trebalo uzeti zdravo za gotovo) ali je problem mnogo složeniji kod raznih dodataka.

Konkretno, RS232 interfejs je najčešće zasnovan na čipu 8250 koji OS/2 jednostavno ne prepoznaje. Jedino rešenje je interfejs sa 16450 koji će ujedno potpuno ravnopravno raditi i pod DOS-om. Sa video karticama je situacija još zamršenija ali na sreću u ovom trenutku je možemo zanemariti — osnovna verzija (bez Presentation Manager-a) radi isključivo u tekst režimu što znači da će odgovarati bilo koji video adapter. OS/2 nije glomazan operativni sistem, tako da se pred hard disk ne postavljaju posebno oštri zahtevi. Pretpostavljamo da bi na nekom najnižem nivou zadovoljio i 30M (40 ms) hard disk, ali se ovakva kombinacija može smatrati nužnim zlom u nedostatku novca. Ozbiljnija korišćenja bi zahtevalo pre svega brži hard disk (recimo 20ms) ali i sa nešto većim kapacitetom — cena 80 ili 80 M diskova neće praviti značajnu razliku u poređenju sa najskupljim elementom — memorijom. OS2 bi teorijski mogao da radi sa nekih 1.5 od 2MB RAM-a, ali sa vrlo lošim efektima. Zato se za osnovnu verziju preporučuje bar 4MB, dok se u kontekstu Presentation Manager'a pominju i dvostruko veće vrednosti. S obzirom na trenutne cene memorijskih čipova (mada su u padu) ispada da memorija zahteva gotovo isti iznos kao i ceo računar pa je ovo verovatno najjači argument protiv OS-a.

Ako ipak prebrodite finansijsku barijeru, sigurni smo da će vam se OS2 izuzetno dopasti. Polako ali sigurno ćete doći u situaciju da će povremeno startovanje računara pod DOS-om biti neprijatno sećanje na prošla vremena.

QUICK BASIC 4.5

Bejzik bez bejzika

Najpopularniji i najrasprostranjeniji programski jezik (ako ste u dilemi, njegovo ime je bejzik) zaista ne može da se pohvali naročitom podrškom našeg časopisa — PC bejzik je poslednji put ozbiljnije predstavljen u „Računarima 35"! Sedamnaestomesečnu bejzik apstinenciju prekidamo prikazom novog *Microsoft*-ovog kompajlera *QuickBASIC 4.50*.

Dejan Ristanović

Iako u svojoj softverskoj ponudi obuhvata brojne programske pakete najrazličitijih namena, *Microsoft* je ipak najpoznatiji kao proizvođač kompajlera i interpretatora programskih jezika — još od starog 4 K Level 1 bejzika. *Microsoft* BASIC je svetski standard za najrasprostranjeniji programski jezik! Stare verzije bejzika su, sasvim prirodno, obuhvatale kakav-takav linijski editor i vršile većinu funkcija operativnog sistema: radilo se, dakle, o integrisanom operativnom sistemu i interpretatoru ugrađenom u ROM. Pojavom CP/M i nešto docnije MS DOS računara, funkcije operativnog sistema su jasno izdvojene, pa su se stekli uslovi za dizajniranje zaokruženih interpretatora i kompajlera — prvi *Microsoft*-ov PC bejzik (popularni BASICA) je i dalje obuhvatao editor (to je ionako bio interpretator) ali su svi docniji kompajleri, sledeći standarde većih kompjuterskih sistema, zahtevali nezavisno unošenje teksta izvornog programa uz pomoć nekog univerzalnog editora, prevođenje u objektni kod i povezivanje (LINK). Na ovaj način su koncipirani i *Microsoft*-ovi kompajleri za fortran, paskal, C itd.

Razvoj informatike, naravno, nikada ne ide u jednom smeru: jednom odbačene koncepcije često se ponovo vraćaju u život. Kompajler odvojen od editora je na velikim sistemima bio prihvatljiv, ali su se vlasnici personalnih računara pokazali kao izbirljive publika: ukupna brzina prevođenja i linkovanja je, usled potrebe da se često pristupa disku, ocenjena kao neodvoljna, trebalo je mnogo „ležijskog" kucanja, bilo je teško pamtiti i ispravljati greške, nezavisni debager je previše mučio početnike...

(Ne)očekivani uspeh *Borland*-ovog *Turbo Pascal*-a koji predstavlja integrisani editor i kompajler naterao je *Microsoft* da preispita svoj korisnički interfejs. Tako su nastale dve paralelne linije programskih jezika — imena bejzik, paskal, C itd. predstavljaju nezavisne kompajlere, dok prefiks *Quick* (npr. *QuickBASIC*, *QuickC*) označava integrisani paket sa editorom, kompajlerom i debagerom. U početku su *Quick* jezici predstavljali podskup „punih jezika" namenjen početnicima i onima koji, plaćajući manji iznos, žele da vide da li će neki jezik odgovarati njihovim potrebama i mogućnostima, ali je tržišna konkurencija učinila svoje — „polujezici" nisu mogli da konkurišu *Borland*-ovim proizvodima, pa je *Microsoft* odlučio da *Quick* varijante praktično izjednači sa punim kompajlerima. Novi *QuickC 2.0* se, na primer, isporučuje na 11 disketa, što je jedva nešto manje od standardnog C-a 5.1, dok se *QuickBASIC 4.5* sa svim približio trenutno aktuelnom BASIC-u 6.0. Ovakvom politikom korisniku je omogućen izbor između integrisane okoline i samostalnog kompajlera bez žrtvovanja karakteristika jezika; vreme izgleda pokazuje da klasični kompajleri postepeno zaostaju za integrisanim paketima.

QuickBASIC je na stranicama „Računara" poslednji put predstavljen kada je aktuelna verzija bila 3.0, a najavljena 4.0. Pokazalo se da *QuickBASIC 4.0* predstavlja veliko unapređenje u odnosu na sve prethodne verzije, ali i na predak ime svoj napredak — krajem 1988. godine na tržištu se pojavio *QuickBASIC 4.5* o kome ovde govorimo. Obzirom na brojne izmene i široku popularnost bejzika, odlučili smo da jezik i kompajler detaljno predstavimo.

Osnove (Quick) bejzika

Premda se zove bejzik i premda će primiti 99% programa pisanih na standardnom bejziku, *QuickBASIC 4.5*, u stvari, predstavlja novi jezik, neku vrstu Module 2 odenute u sintaksu bejzika. Novitati se pokazuju već kod tipova podataka: umesto celih i racionalnih brojeva i stringova, *QuickBASIC* podržava šesnaestobitne cele brojeve (*integer* iz intervala −32768..+32767), tridesetdvobitne cele brojeve (*long integer* iz intervala −2 147 483 648..+2 147 483 647), racionalne brojeve (sedam tačnih cifara, apsolutne vrednosti u intervalu [1.40129E−45, 3.40282E+38]), dvostruko tačne racionalne brojeve (petnaest od šesnaest tačnih cifara, apsolutne vrednosti u intervalu [4.940E−324, 1.7976E+308]) i stringove fiksne i promenljive dužine koji mogu da zapamte čak od 32767 slova.

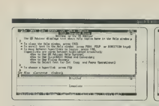
Dok celi, racionalni i dvostruko tačni brojevi predstavljaju standardne IEEE formate koje podržava 80×87 aritmetički koprocesor (*Microsoft* je konačno odustao od sopstvene reprezentacije brojeva premda je uvođenjem odgovarajućih kompajlerskih opcija omogućio i prevođenje ranije napisanih bejzik programa), stringovi su specijalitet *QuickBASIC*-a — vrlo je teško naći programski jezik koji podržava stringove duže od 255 znakova! String osim toga može da bude fiksne ili promenljive dužine — pri radu sa nekim „normalnim" bejzikom operišemo isključivo sa promenljivim stringovima iako da smo u mogućnosti da ih po volji produžavamo ili skraćujemo — funkcija LEN će uvek vratiti broj slova u stringu. Dužina fiksnih stringova se određuje u trenutku dimenzionisanja i uvek je jednaka — LEN (string) će uvek vratiti isti broj dok će, po dodeljivanju kraće reči, ostatak biti dopunjen blankovima. Stringovi fiksne dužine će se u praksi retke koristiti — pogodni su jedino za red sa *random access* datotekama koje su sastavljene od slogova jednakih dužine.

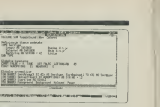
Relativno obilje tipova podataka (nedostaju jedino logičke *(Boolean)* vrednosti koje se moraju simulirati preko celih brojeva) zahteva brojne deklaracije na početku svakog programa. Obavezno deklarisanje promenljivih sa druge strane nije baš u skladu sa koncepcijom bejzika i ne obezbeđuje prenošenje postojećih programa, pa je *Microsoft* napravio određene kompromise — zadržana je implicitna konvencija po kojoj je svaka nedeklarisana promenljiva racionalna i uvedeni sufiksi % & ! # $ (procenat, emperhend, uzvičnik, „taraba" i dolar) koji respektivno predstavljaju cele *(integer)*, dvostruko tačne cele *(long)*, racionalne *(single)* i dvostruko tačne racionalne *(double)* brojeve, odnosno stringove. Ukoliko je ovo teško za pamćenje, možemo da koristimo staru deklaraciju DEFtip (DEFINT, DEFLNG, DEFSNG, DEFDBL ili DEFSTR) ili da, po novoj metodologiji koja najviše odgovara *QuickBASIC*-u, izvršimo nešto poput *DIM proba AS INTEGER* ili *DIM tekst AS STRING *20*. Službena reč AT očito predstavlja zamenu za paskal programerima dobro poznatu dvotačku koju smeštamo između imena promenljive i njenog tipa.

QuickBASIC podržava i slogove: službena reč TYPE je zamena za paskalsko RECORD, dok se END zamenjuje sa END TYPE — sve između je potpuno analogno sa paskalom, s tim što je dvotačka zamenjena sa AS. Sve u svemu, u domenu tipova podataka paskal ima svega dve prednosti nad *QuickBASIC*-om: promenljive tipa *Boolean* i mogućnost definisanja sopstvenih poddomenskih tipova (npr. type boje = (plava, žuta, crvena)). Prva razlika je umereno značajna dok druga, iako naoko važna, ne predstavlja poseban argument: lepo je što u paskalu možemo da definišemo tipove koji su nam potrebni, ali sa njima ne možemo da uradimo bog zna šta: nije, na primer, moguće njihovo ispisivanje ili vraćanje iz funkcijskog potprograma. Ukoliko bi se sa druge strane *QuickBASIC* poredio sa nekom modulom 2

Quick Basic

Verzija
4.50, 28. septembar 1988.
Namena
Bejzik kompajler
Sadržaj paketa
Pet XT disketa, tri priručnika
Hardversko-softversko okruženje IBM PC/XT/AT ili kompatibilan, 512 K RAM-a, flopi ili hard disk, DOS 2.10 ili noviji
Zauzet prostor na disku
Oko 1.7 megabajta
Ulaz
Tastatura, miš
Izlaz — ekran
CGA, EGA, VGA i (delimično) Hercules
Dokumentacija
Programming in BASIC — 460 A5
BASIC Language Reference — 600 A5
Learning and Using Microsoft QuickBASIC (400 A5)
Proizvođač
Microsoft Corporation
13221 SE 26 th. Suite L
Bellevue, WA 98005
U.S.A.





(što je prilično jeretička ali ne i nerazumna misao) nemogućnost definisanja sopstvenih tipova bi mogla da predstavlja značajnu prednost.

Dok se o raspoloživim tipovima podataka može duže diskutovati, superiornost *QuickBASIC*-a u odnosu na paskal u oblasti kontrolnih struktura je neosporna. Tu je, pre svega, klasična višelinijska IF ... THEN ... ELSEIF ... ELSE ... END IF konstrukcija koja se, zahvaljujući očuvanju jednolinijskih IF-ova, savršeno uklapa kako u stare verzije bejzika tako i u strukturirane trendove. Izbor obezbeđuje i SELECT CASE sa svim dodacima koji uključuju i ELSE.

Univerzalna petlja *QuickBASIC*-a zove se DO ... LOOP i zamenjuje ne samo WHILE i REPEAT nego i sve druge petlje koje nam padaju na pamet — možemo da napišemo DO WHILE uslov ... LOOP, DO UNTIL uslov ... LOOP, DO ... LOOP WHILE uslov i DO ... UNTIL uslov. Samo DO ... LOOP predstavlja beskonačnu petlju iz koje se po potrebi izlazi komandom EXIT — ista komanda može da se koristi i u svim drugim petljama tako da se, na primer, u slučaju greške može „iskočiti" iz petlje pre ispunjenja uslova za njen regularan završetak. Specijalitet *QuickBASIC*-a je komanda EXIT FOR koja obezbeđuje iskakanje iz brojačke (FOR ... NEXT) petlje, efekat koji je zastupljen u veoma malom broju programskih jezika. Samo se po sebi razume da je komanda GOTO (labele mogu da budu kako brojevi tako i alfanumerici) i „stara petlja" WHILE ... WEND zadržana, ali će ova struktura pored noviteta koje je *QuickBASIC* ponudio verovatno ubrzo pasti u zaborav.

QuickBASIC stimuliše programera na maksimalno modularni koncept rada — umesto „antičkog" GOSUB preporučuje se pozivanje procedure sa CALL ime (argumenti) i upotreba funkcijskih potprograma koji se pozivaju prostim navođenjem imena. Sam radni deo potprograma započinje sa SUB ime (odnosno FUNCTION ime) i završava se sa END SUB (END FUNCTION). Argumenti se standardno prenose po imenu, što znači da procedura može promeniti svaku od prenetih joj vrednosti — ako želimo da ovo izbegnemo, dovoljno je da umesto stvarnog argumenta unesemo neki izraz (poslužiće i (A) ili A+0) što je znatno nepogodnije od paskala kod koga se za prenošenje po imenu ili po vrednosti opredeljujemo u trenutku pisanja procedure, a ne u trenutku njenog pozivanja. Prenošenje matrica je rešeno veoma fleksibilno — procedura se može napisati tako da operiše sa matricama proizvoljnih dimenzija, pri čemu su obezbeđene funkcije koje vraćaju vrednost najmanjeg i najvećeg indeksa prenesene matrice.

U okviru procedura ili funkcija mogu se koristiti lokalne promenljive koje nemaju nikakve veze sa istoimenim vrednostima glavnog programa — A u proceduri nije isto što i A u programu! Dodavanjem deklaracije STATIC može se obezbediti da se generisana vrednost lokalne promenljive ili svih lokalnih promenljivih očuva i pri sledećem ulasku u proceduru. Ukoliko se ukaže potreba za prenošenjem parametara u potprograma bez upotrebe liste, *QuickBASIC* obezbeđuje COMMON zone posredstvom kojih se vrednosti mogu prenositi i drugim programima koji se startuju sa RUN.

Sastavljanje programa od procedura upisanih u različite datoteke nameće kompajleru određene probleme — kako znati da li je PROBA ime promenljive ili ime funkcije? Ukoliko za pisanje programa koristite sam *QuickBASIC*, pre svakog snimanja izvornog programa na početak datoteke biva ugrađen odgovarajući broj DECLARE redova u kojima figuriše ime svake procedure i funkcije koja se nalazi u modulu. Ukoliko koristite neki drugi editor ili imate procedure i funkcije u drugim modulima, potrebne DECLARE redove morate otkucati sami. U okviru dokumentacije se savetuje da uz svaki skup korisnih procedura formirate prateću datoteku sa njihovim deklaracijama i da onda tu datoteku (uz pomoć $INCLUDE) ubacujete u svaki program koji će te procedure koristiti. Mora se primetiti da je rešenje primenjeno u *Turbo Pascal*-u pogodnije jer je lista deklaracija (INTERFACE sekcija) integralni deo svakog modula što značajno olakšava povezivanje. *Turbo Pascal* osim toga omogućava svakom modulu da ima sopstvenu inicijalizacionu sekciju koja će se izvršiti pre nego što se kontrola prepusti glavnom programu i tako pripremiti neke od parametara koji su modulu potrebni. U *QuickBASIC*-u se ovakva sekcija (ako je i navedemo) uopšte ne izvršava, što znači da glavni program na početku rada mora pozvati procedure koje inicijalizuju podređene module — garantujemo da ćete pri sledećoj upotrebi modula zaboraviti da ga inicijalizujete!

Što se rada sa datotekama tiče, u *QuickBASIC*-u nema šta nema. Možete se pre svega opredeliti za klasične sekvencijalne datoteke sa tekstom čiji redovi različitih dužina predstavljaju pojedinačne slogove; polja tih slogova se razdvajaju zarezom, tabulatorom ili time što je svako polje „zaokruženo" navodnicima. Tu su zatim datoteke sa slobodnim pristupom (*random access*) čiji slogovi imaju fiksnu dužinu i koje možete kreirati, čitati, produžavati i ispravljati. Za sistemske potrebe su najznačajnije „datoteke bajtova" (sadržaj je bilo šta) kojima možemo slobodno pristupati počevši od bilo kog mesta primenom komande SEEK. Da bi se svi ovi tipovi opslužili i da bi se održala kompatibilnost sa ranijim verzijama bejzika, uveden je mali milion procedura i funkcija kao i odgovarajući broj parametara stare komande OPEN — *QuickBASIC* zaista ima potencijale za najrazličitije obrade podataka, ali će privikavanje na njegovu sintaksu zahtevati određeno listanje dokumentacije i (ne)određeno vreme.

QuickBASIC je najzad opremljen brojnim naredbama koje rade sa grafikom, pri čemu su podržane „zvanične" kartice CGA, EGA i VGA; podrška kod nas ubedljivo najpopularnijem Herculesu se svodi na rezidentni program MSHERC (u verziji 4.0 QBHERC) posle čijeg će startovanje većine grafičkih programa raditi ... nekako, to jest prilično sporo.

Radna okolina

Pošto se upoznamo sa svim pomenutim i brojnim nepomenutim lepotama *QuickBASIC*-a, poželećemo da upoznamo i način na koji se one saopštavaju računaru. Datoteke sa programom se, pre svega, može kreirati primenom bilo kog programerskog tekst editora i prevesti sa BC ime. Ovakav rad naravno nije u skladu sa filozofijom *Microsoft*-ovih *Quick* jezika, pa se u većini slučajeva koristi integrisana radna okolina zvana QB.

Editor ugrađen u QB je relativno standardan i dobro prilagođen potrebama bejzik programera — kontrolne sekvence za obavljanje složenijih operacija podsećaju na *WordStar* ali se mogu zameniti prostim pokretima miša. Pošto svaki programer nema miša i kompjutersko pamćenje, sekvence duže od jednog znaka su propraćene dnevnim HELP-om koji će, kada na primer pritisnemo Ctrl+K, ispisati značenje

Šta ima novo

Iskusniji korisnici QuickBASIC-a 4.0 će svakako želeti da znaju kakve je novitete nova verzija ponudila. Što se tiče jezika, brzine prevođenja i rezultujućeg izvršnog koda, razlike su beznačajne — sitnu novost zvanu ON UEVENT neće svako koristiti. Radni prostor je unekoliko unapređen uvođenjem novog Options menija u okviru koga definišemo direktorijume za datoteke raznih tipova i eventualno se opredeljujemo za kraće menije (samo najvažnije opcije). Tu je i nova Instant Watch opcija koja obezbeđuje interaktivno izračunavanje vrednosti izraza u toku rada sa kompajlerom.

Glavna novost je izuzetno opširni i dobro organizovani interaktivni HELP koji omogućava programeru da se podseti sintakse neke naredbe odnosno funkcije ili da brzo pregleda podatke o nekoj komandi editora ili radnog prostora. Kretanje po HELP biblioteci je, zahvaljujući doslednoj upotrebi miša, vrlo racionalno i funkcionalno, dok je korisnicima čiji je hard disk uvek prepun omogućeno aktiviranje skraćenog HELP-a ili potpuno ukidanje računarske pomoći.

Za i protiv

Hvalimo

1. Brojne ekstenzije
2. Savršene kontrolne strukture
3. Kontrolu grešaka i događaja u sistemu
4. Duge stringove i nizove
5. HELP i izvanrednu literaturu

Kritikujemo

1. Probleme sa prenošenjem argumenata po vrednosti
2. Slabu podršku Hercules kartice
3. Nemogućnost definisanja komandi editora

svakog od mogućih nastavaka ove sekvence lako ćemo odmah znati da je Ctrl+K B početak bloka a Ctrl+K K njegov kraj.

Pošto je program unesen i snimljen na disk kucamo Alt+R i započinjemo njegovo prevođenje i (ukoliko ne bude grešaka) izvršavanje. Primenom integrisanog dibagera izvršavanje možemo vršiti „korak po korak" uz automatsko ili dirigovano ispisivanje i promenu vrednosti „opasnih" promenljivih. *QuickBASIC 4.5* je na taj način uklonio jednu od velikih razlika između kompajlera i interpretatora — ukoliko nam je pri radu potrebno da izračunamo koliko je 2+2 ili COS (PI/7) otkucaćemo jednu komandnu liniju i pročitati rezultat — *SideKick* gotovo da nije potreban!

Kada je razvoj programa završen poželećemo da proizvedemo nezavisni EXE fajl koji će se izvršavati i na mašinama čiji vlasnici nisu prešli na *QuickBASIC*. Ova operacija je relativno osetljiva jer kompajleru treba saopštiti koji sve moduli ulaze u program (svi se moduli upisuju u editor) i po mogućnosti isključiti iz njih one procedure koje program ne koristi — *Borland*-ov linker bi potpuno samostalno obavio ovo filtriranje kada bi bio kompatibilan *QuickBASIC*-om. Treba se osim toga opredeliti i za biblioteku sa kojom se program linkuje — glavni parametri je postojanje aritmetičkog koprocesora i želja da bejzik program bude potpuno samostalan ili da je za njegovo izvršavanje potreban program BRUN45. U globalu ćemo ako prevodimo za sebe proizvoditi „ne baš nezavisne" EXE fajlove (znatno su kraći) a za tržište samostalne aplikacije. Sama „proizvodnja EXE programa" je prilično spora (*Turbo Pascal* obavi posao za trenutak) ali je s obzirom da se radi o ređe potrebnoj operaciji sve u prihvatljivim granicama.

Autori *QuickBASIC*-a su vodili računa i o tvorcima softverskih poluproizvoda tj. biblioteka potprograma koji se prodaju profesionalnim korisnicima koji imaju posebne potrebe (npr. biblioteka za rad sa matricama, upravljanje Epson kompatibilnim štampačem, komunikacijom sa dBASE-om i tome slično). Ukoliko ne želite da distribuirate izvorni kod možete se opredeliti za standardni OBJ fajl ili što je mnogo pogodnije *Quick LIB* biblioteku prevedenih programa koji su pisani na raznim *Microsoft*-ovim jezicima. Distribuiranje *Quick LIB*-a ne uklanja potrebu za priključivanjem odgovarajuće LIB datoteke jer se bez nje ne može kreirati samostalna izvršna aplikacija.

Zanimljivo je da se *Microsoft* popul *Borland*-a nije potrudio da obezbedi program za prevođenje biblioteka koje je kreirao *QuickBASIC 4.0* u format koji odgovara *QuickBASIC*-u 4.5 — programe morate ponovo prevoditi što i nije neki problem ako vam je na raspolaganju izvorni kod. Kupcima pomenutih softverskih poluproizvoda ostaje samo da platе novu verziju ili rezervišu prostor na disku za starи verziju *QuickBASIC*-a. Ovakve nebriga za redovne kupce kako nam se čini ne odgovara renomeu jednog *Microsoft*-a ili *Borland*-a.

Sve ima svoje granice pa tako i mogućnosti *QuickBASIC*-a. Mora se međutim reći da su granice više nego velikodušne: string kao što smo videli može da ima najviše 32767 znakove (na *Turbo Pascal*-u 255), memorijski prostor koji svaki niz zauzima je ograničen na 64 K (na *Turbo Pascal*-u ako nećete da koristite trikove imate 64 K za sve podatke), isto ograničenje važi i za svaki pojedini programski modul. Ostala ograničenja su manje važna: najduže ime promenljive može da ima 40 znakove, matrica može imati najviše osam dimenzija pri čemu brojač po svakoj od dimenzija mora biti između −32768 i +32767, procedure ne sme imati više od 60 argumenata, dužina sloga datoteke ne sme preći 32 K. Kada se pomirite sa činjenicom da se na PC-ju ipak ne može pristupati kompletnoj memoriji i saživite sa segmentima od po 64 K *QuickBASIC* će za vas biti jezik bez granica!

Dokumentacija programskog paketa je više nego kompletna — dobijate generalnu knjigu zvanu *Programming in BASIC* u okviru koje su objašnjene osnove *QuickBASIC*-a i pobrojane brojne ekstenzije, *BASIC Language Reference* u okviru koga je detaljno opisana svaka naredba i funkcija bejzika i *Learning and Using Microsoft QuickBASIC* sa detaljnim opisom integrisanog radnog prostora i debagera. Kvalitet dokumentacije je izuzetno visok, činjenice su propraćene brojnim primerima, dok sadržaj i indeks predstavljaju pravo remek delo — najpre sadržaj sadržaja pa onda detaljan sadržaj i onda sve to ponovljeno na početku svakog poglavlja. Učenje bejzika uz pomoć *Microsoft*-ove literature predstavlja pravo zadovoljstvo.

Jedan od noviteta *QuickBASIC*-a 4.5 je izuzetno opsežan HELP o kome govorimo u okviru posebnog boksa. HELP je praćen tutorijalom koji pomalo podseća na izvanredni *Excel*: kroz primere i testove PC će vas za svega desetak minuta naučiti da koristite osnovne komponente radnog prostora *QuickBASIC*-a.

QuickBASIC je sve u svemu izvanredna programerska alatka koja ne žrtvujući jednostavnost bejzika promoviše sve elemente savremenog struktuiranog programiranja približavajući tako pisanje većih programskih sistema relativnim početnicima. Upoređen sa neposrednim konkurentom *Turbo Pascal*-om *QuickBASIC* nudi određene prednosti i ima određene mane — prednost su stringovi dugi 32767 znakova, neograničen segment podataka, mnogo veći broj ugrađenih funkcija i procedura, brojne kontrolne strukture, maksimalno pojednostavljen tretman grešaka (ON ERROR, ON KEY itd) i generisanje odnosno prihvatanje OBJ datoteka koje ipak predstavljaju univerzalni standard. *Turbo Pascal* brže prevodi i generiše brži kod (*QuickBASIC* programi su i dalje do 50% sporiji od *Turbo BASIC*-a i *Turbo Pascal*-a), omogućava definisanje editorskih komandi, ima bolji debager, brojnije grafičke funkcije (podržan je i Herculesj i nekako je dosledniji — da bi sačuvao kompatibilnost sa ranijim verzijama bejzika *Microsoft* je morao da omogući da se ista stvar uradi na dva ili tri različita načina što početniku može izgledati zbunjujuće. Iako je izbor između *QuickBASIC*-a, *Turbo BASIC*-a i *Turbo Pascal*-a veoma težak čini nam se da je stvar ipak jednostavna — ni za jednog bejzik programera koji se odlučio za *QuickBASIC* se ne može reći da je pogrešio!

Executive Type Clasic Fontpack

Fontovi bez muke

VS Software je jedna od prvih firmi koju smo upoznali kada su laseri zauzeli mesto na našim stolovima. Njihov *FontGen IV* editor bit mapiranih laserskih fontova je verovatno i dalje nezamenljiva alatka za sitne radove: iako program pati od mnogih nedostataka, mnogi ga i dalje koriste za sitnije intervencije (tipično dodavanje akcenata) u fontu. S obzirom da je *FontGen IV* relativno star proizvod (naravno, mereno softverskim godinama) i da nismo čuli za nove verzije programa, na VS Software smo potpuno zaboravili. Međutim . . .

Zoran Živolić



pismo koje je stiglo u redakciju pre nekoliko meseci pokazalo je da je VS i dalje živ i zdrav i da je u potpunosti nastavio osnovnu razvojnu koncepciju kompletiranja podrške laserskim štampačima i stonom izdavaštvu uopšte. Proizvodi koji su potom stigli na testiranje pokazali su takođe da VS ima puno svežih ideja. U ovom broju ćemo kao mali uvod u sledeće prikaze proizvoda ove firme pokazati font paket koji nam je stavljen na raspolaganje — *Executive Type Classic FontPak*.

Font paketi u VS su u tradicionalnom obliku — gotovi SFP fontovi u kondenzovanom formatu. Iako bi se moglo reći da posle *Bitstream*-a i fleksibilne varijante raspakivanja fontova po želji ovakav način deluje zastarelo, tvrdnja je u potpunosti pogrešna ako su ispunjena tri uslova — (1) izbor veličina zadovoljava potrebe, (2) automatizovana instalacija bez potrebe za poznavanjem *Venturinog* font mehanizma i najvažnije (3) ako je kvalitet fontova odličan. Ovaj poslednji razlog može biti i najjači pogotovo što se firmi pruža prilika da na gotovoj bit mapi vrše dodatne optimizacije kako bi aproksimacija pisma bila najbolja moguća.

Executive Type Clasics Fontpak se isporučuje na 7 disketa i sadrži sledeća pisma: VS Bookman, VS Century, ITC Souvenir, CG Omega, ITC Avant Garde Gothic.

S obzirom da se radi o pismima kojima se uglavnom sloga osnovni tekst, izbor veličina (slika 1) je prilagođen ovim potrebama i može se smatrati zadovoljavajućim.

Uz svaki paket se dobija i instalaciona disketa sa koje se pokretanjem programa INSTALL fontovi prenose u *Venturinu* ili okolinu drugih programa. Na slici 2 je data uvodna maska sa koje se može videti koje sve programe VS podržava. Naravno, najviše nas je interesovala Ventura pa nas je nakon izbora ove opcije sačekalo malo iznenađenje — program je „protrčao" po našem hard disku i samouverano prijavio da se kod nas Ventura nalazi u direktorijumu E:/VENTURA a da su fontovi izolovani u E:/VPHPFONTS i tražio samo potvrdu o instalaciji. Naravno, programu nije teško da pronađe ovaj podatak ali činjenica da o ovome niko pre VS nije razmišljao govori više u prilog VS-u nego što je sam značaj ovog trivijalnog primera.

Instalacija teče na uobičajen način — redom se ubacuju diskete a instalacioni program prepoznaje tip podataka na njima i preduzima odgovarajuće akcije — raspakuje fontove, prenosi WID tablice itd. Nakon instalacije na disku će se naći sve što je potrebno za započinjanje posla. Svako od pisama je praćeno odgovarajućim ekranskim fontom i WID tablicom.

Sa ekranskim fontovima smo imali problema. Uz svako pismo se nalazio samo po jedan ekranski font nepoznate veličine koja je Ventura bila primorana da generiše sve veličine što ne daje dobre rezultate. Nedostatak ekranskih fontova za ova pisma nije ozbiljan problem — verovatno je preovladala želja da se Ventura ne opterećuje velikim brojem ekranskih fontova pogotovo što se i ne mogu adekvatno prikazati na ekranu. U ovim situacijama se koristi aproksimacija tajmsom i helvetikom, ali VS instalacioni program ne pruža nikakave mogućnosti da se izabere ova opcija pa je ovo ozbiljan nedostatak.

Kvalitet fontova je svakako najinteresantnije pitanje. Sva pisma deluju veoma ujednačeno sa vrlo čistim izgledom i bez mestimičnih grubljenja oblika. Čak i neki veoma delikatni oblici u light verziji Souvenir-a su odlično izvedeni — gotovo da nema načina da na laserskoj rezoluciji izgledaju bolje.

Ukupan utisak kvari nekoliko grešaka — u pomenutom Souvenir-u malo slovo t ima nekih nedoslednosti u oblicima koja su vrlo čudne a slično se dešava i na nekim bold slovima Bookman-a. S obzirom na ukupan izgled pisma koji je odličan, skloni smo da ove greške pripišemo pre programu za raspakivanje fontova nego dizajneru. Fontovi se isporučuju u dve varijante kodnog standarda — USA ASCII (7 bita) i prošireni ASCII sa slovima evropskih azbuka. Naših slova nema, ali sa FontGen-om ili sličnim programom nije previše napomo uneti odgovarajuće izmene.

```
VS Software Tool Kit & Font Installation  (c)Copyright VS Software 1987 88
        for H-P LaserJet Plus  500+  Series II  2000 or compatables

        Enter an  X  fallowing the name of the Word Processing
          or Desktop Publishing P-ckage you will be using

              MicroSoft WORD
              WordPerfect VERSION 3 OR 4
              WordPerfect VERSION 5 0
              Aldus PageMaker
              Ventura Publisher
              XYQUEST XyWrite III
              MicroPro Word Star 2000 Rel 3
              Polaris PrintMerge
              Spellbinder Desktop Publisher
              Softlab LaserType
              Printworks
              MicroSoft Windows
              Other Word Processor

     Use (↑↓) Arrow Keys  (Enter) or (Tab) to position cursor
                         (Esc) to return to DOS
```

Korisna adresa
VS SOFTWARE
A Division of VideoSoft, Inc.
P.O. Box 6158 Little Rock, AR 72216
501/376 2083

Iz riznice „Adinog kruga"

Zov fraktala

Među najnovijim disketama iz ponude „Adinog kruga" svatko će naći po nešto za sebe, ali je sigurno da će „Monsters", program za generiranje fraktala, privući pažnju svih, jer ljepoti fraktala je jednako teško odoljeti kao i zovu sirene.

Zoran Cvijetić

ADK 156

Ova disketa je nastavak niza namijenjenog C programerima. Sadrži skup funkcija za rad sa karakterima i njihovim nizovima, redirekcioniranje ulaza i izlaza, automatsko proširivanje imena datoteka zadanih sa 'wild' karakterima u komandnoj liniji i sortiranje datoteka po ključu. Sve su pisane za prevodioce na CP/M u

Tu je i program za traženje uzorka u nizu datoteka.

ADK 159

„E" je editor 'full screen' tipa namijenjen programerima (što ne znači da ga i drugi ne mogu koristiti). Broj funkcija kojima raspolaže je velik, a mi ćemo spomenuti samo automatsko poravnavanje namijenjeno strukturama, mogućnost horizontalnog pomicanja bloka radi poravnavanja i undo funkciju koja operira sa 40 zadnjih promjena.

Izvorni kod je dan (potreban je CP/M BDS C prevodioc, verzija 1.50), što otvara velike mogućnosti prilagođavanja „E"-a ličnom ukusu i potrebama.

ADK 160

Crtanje fraktala, tih čudesnih krivulja magične privlačnosti, predstavlja tvrd orah za svaki personalni računar zbog velikog broja operacija koje je potrebno obaviti. Bez obzira što se vrijeme potrebno za generiranje slike zadovoljavajuće kvalitete mjeri desetinama minuta pa čak i satima, ne poznajemo nijednog zaljubljenika u računare koji je uspio odoliti porivu da iscrta makar jednu takvu sliku. Po tome svaki od njih želi što više detalja u što više boja, što eksponencijalno produžava vrijeme crtanja. Najbolje ali i najegzotičnije rješenje ovog problema (crtanje što kvalitetnije slike fraktala u što kraćem vremenu) smo vidjeli u „BYTE"-u kada je Steve Ciarcia, njihov hardverski čarobnjak, napravio višeprocesorski računar kojem je dao ime Mandelbrotova mašina i kojem je prvi zadatak bio, već pogađate, da generira Mandelbrotova skupove.

Manje egzotično rješenje od Ciarcijnog ali još uvijek zadovoljavajuće i, što je za prosječnog Jugoslovena najvažnije, ekonomski najpovoljnije je nabavka programa „Monsters" Ciala Pezdirca. „Monsters" je program za oblikovanje regularnih fraktala i krivulja za popunjavanje prostora koji svoj posao obavlja nevjerojatno brzo (posljedica metode koju koristi) a pri tom podržava niz grafičkih kartica (HERCULES, EGA i VGA). Ovo je shvatljivo kada se zna da je pisan u „Turbo Pascal"-u 4.0. Izvorni kod je priložen.

Metoda koja se koristi u „Monsters"-u je opisana u „BYTE"-u (avgust 1987.) i njena ideja vodilja je da se svaka krivulja za popunjavanje prostora može predstaviti specifičnim „genetskim kodom" koji se zadaje tabelarno.

„Monsters" posjeduje ljubazan korisnički interfejs pa je definiranje krivulja izuzetno jednostavno, a priložena je i biblioteka sa oko 20 predefiniranih krivulja (Hilbertove, Mandelbrotova, Gosperjeva, Kochove, Sierpinskoge i druge). Podaci za predefinirane krivulje su preuzeti iz spomenutog članka.

Nakon višesatnog bavljenja „Monsters"-om jedini odgovarajući opis koji smo našli za njega je bio „Zadivljujuće!"

ADK 162

Izvorni kod verzije 1.1 legendarnog „Small C"-a (autor Ron Cain) je revidiran, pa zato obratite pažnju na ovu disketu. Kod je tačno onakav kakav je objavljen u časopisu „Dr. Dobb's Journal of Computer Calisthenics & Orthodontia" (broj 45).

ADK 163

C prevodilac bez biblioteka je isto što i pas bez zubiju: laje, ali ne ujede. Ova i sljedeće dvije diskete oštre zube vašem C-u jer su pune biblioteka razne namjene.

„SMLATS — Steve's C Library" je biblioteka namijenjena „small" memorijskom modelu „Lattice C"-a. Namjene funkcija je toliko raznolika da je grupiranje za potrebe predstavljanja u članku praktično nemoguće (trebali bi opisati svaku funkciju ponaosob). Zato vam samo preporučujemo da poklonite pažnju ovoj biblioteci jer smo sigurni da u njoj ima za svakoga po nešto.

Izvorni kod, nažalost, nije priložen, ali se može dobiti od autora.

ADK 164

COMMISC.C sadrži niz funkcija za „Lattice C" namijenjenih interaptski kontroliranom I/O-u na PC-u.

MATH.C je niz matematičkih funkcija za „Lattice" ili „Microsoft" C prevodioce. Služe za računanje drugog korijena, 2^x (korištenjem polinomske aproksimacije), e^x, 10^x, sinusa, arkus sinusa, arkus tangensa, logaritma po bazi 10, prirodnog logaritma, eksponenciranje i računanje modulo funkcije.

Računanje se vrši sa polinomskim ili racionalnim aproksimacijama bez korištenja matematičkog koprocesora. Tačnost je na prvih 9 do 10 znamenki.

Funkcije za čišćenje ekrana, mijenjanje video moda, pomicanje kursora na željenu poziciju, očitavanje pozicije kursora, određivanje aktivne video stranice, iscrtavanje tačke na proizvoljnoj poziciji ekrana u grafičkom modu kao i određivanje koje je boje tačka na nekoj poziciji ekrana u grafičkom modu se mogu naći u SCRNMISC.C. Sve su namijenjene verziji 2.12 „Lattice C" prevodioca.

WINDOWS.LIB sadrži „C Windows" biblioteku funkcija za preklapajuće prozore. S njom korisnici „Lattice C"-a 2.12 mogu otvarati prozore (od 255 njih istovremeno), premještati ih, zatvarati, reaktivirati, brisati i skrolirati. Pri tome se svaki od prozora tretira kao odvojeni virtuelni ekran.

Izvorni kod nije priložen, ali se može dobiti od autora.

CUT.C i PASTE.C su programi za istoimene UNIX komande.

ADK 165

Datoteke CLOGS.C, CTRIG.C i FLOAT44.DOC sa ove diskete sadrže niz matematičkih funkcija (logaritamske, trigonometrijske i funkcije za rad sa brojevima u tekućem zarezu). U dokumentaciji ne piše kojem prevodiocu su namijenjene, ali vas to teoretski ne bi trebalo smetati jer C je izvikan kao prenosiv jezik, zar ne? Ukoliko budete imali probleme oko prenošenja, razmislite da li ste zavedeni blagoglagoljivošću advokata C-a načinili pogrešan izbor!

Na kraju spomenimo da se grupi jugoslovenskih pisaca softvera u javnom vlasništvu pridružio Vojko Kalan sa programima na disketi ADK 161.

Tehnički detalji

Namjena: Razno
Sadržaj paketa: Samo diskete
Broj disketa: Testirali smo diskete sa rednim brojem ADK 156, 159, 160, 162, 163, 164 i 165
Tip disketa: Diskete su veličine 5.25" u standardnom PC formatu zapisa
Dokumentacija: Priručnik ne postoji u štampanoj formi, već je u obliku datoteke na disketi
Okruženje: IBM PC/XT/AT, PS/2 ili kompatibilan računar. Tvrdi disk nije potreban. Potrebna verzija DOS-a 2.00 ili viša.
Proizvođač: Detaljne informacije o načinu učlanjivanja i naručivanja možete dobiti na adresi:

MIKRO ADA
(ZA ADIN KRUG)
Cankarjeva 10 b
61000 LJUBLJANA
Tel. (061) 219/125

Cijena: Cijena jedne diskete sa programima ovisi o tome da li ste član „Adinog kruga" ili ne, da li ste privatno lice ili društvena organizacija i o tome koliko disketa naručujete odjednom. Raspon je od 15000 do 35000 dinara po komadu (prema cjenovniku iz aprila 1989). U cijenu nije uračunata poštarina. Upozorenje: cijene se mijenjaju svakih 2 mjeseca!


INSTITUT Mihajlo Pupin
tim 030
BESKRAJNI PROSTORI RAČUNARSTVA
IMP
INSTITUT „M. PUPIN"
RJ Računarstvo
Beograd, Volgina br. 15
tel.: 011/772-876
telex: 11584 YU imp. bg

Kućni i drugi poslovi

Većinu programa koje svakodnevno koristimo možemo, prema funkciji koju vrše, lako klasifikovati kao tekst-procesore, grafičke programe, DTP-programe, baze podataka, .ACC-programe, igre, itd. Svaka od tih klasa se može dalje podeliti na podgrupe, pa tako imamo klasične i grafičke tekst-procesore, pikselorijentisane i objektno orijentisane grafičke programe, CAD-programe, programe za animaciju, itd. Međutim, postoji jedna vrlo široka i šarolika grupa uslužnih programa nazvana Utility.

Marko Kirić

Naziv utility je usvojen jednostavno zato što programi koji se svrstavaju u ovu grupu nemaju nijednu drugu zajedničku karakteristiku koja bi podjednako važila za sve. Ovde jednostavno spadaju svi programi koji obavljaju takozvane „kućne poslove" kao i pomoćni programi napravljeni kao dopuna drugim programima ili samom operativnom sistemu. Gotovo je nemoguće napraviti kompletan spisak takvih programa i njihovih mogućnosti i namena, što vrlo često dovodi do toga da se naročito početnici osećaju izgubljeni pred tim mnoštvom i počnu da akupljaju sve od reda u nadi da će jednog dana moći da razaberu šta je šta i koje od tih programa treba sačuvati i koristiti a koje ne.

Da bi situacija bila još konfuznija, takvi programi pored slikovitih a vrlo često i kriptičnih imena imaju i veoma različite dužine, tako da praktično ne postoji nikakav kriterijum po kome bi se mogla proceniti upotrebljivost nekog od ovih programa izuzev preporuka drugih korisnika koji su ih već isprobali. Sigurno je samo to da se bez ovakvih programa može raditi, ali pravilan izbor i upotreba jednog ograničenog broja može znatno da pojednostavi i ubrza rad i to do te mere da ćete često čuti iskusnije korisnike kako kažu da bez tog i tog programa jednostavno ne mogu da zamisle bilo kakav ozbiljan rad. To je, naravno, kao i sve drugo što je vezano za ljude i ljudsku prirodu, stvar navike i ličnih sklonosti. Svrha ovog članka je da vam, pre nego što takve navike steknete, pruži makar delimičan pregled najčešće korišćenih programa ove vrste i njihovih mogućnosti i da vam olakša izbor.

Iako se radi o krajnje raznorodnim programima, jedan veći broj je moguće, prema nameni, klasifikovati po grupama, dok jedan manji broj obavlja tako specifične poslove da jednostavno ne mogu da se uporede ni sa jednim drugim programom ove grupe.

Disk editori

Programi koji spadaju u ovu grupu su, uz RAM-disk programe i programe za formatiranje i kopiranje, svakako najčešće i najšire korišćeni programi na računarima ST serije. Kao što im i samo ime govori, osnovna namena ovih programa je direktan prilaz sadržaju diska, i to sektor po sektor, pri čemu je moguće menjati pojedinačne bajtove unutar sektora. Drugim rečima, omogućuju svakovrsno čačkanje i čeprkanje po diskovima, i to počev od prevođenja poruka pojedinih programa, popravljanje greškom izbrisanih programa ili podataka, pa sve do raznih eksperimenata i uopšte svega onoga što nazivamo hakerisanjem. Ova vrsta zanimacije je zajednički hobi i neka vrsta kolektivnog zaštitnog znaka svih AtariSTa, tako da bi se teško mogao naći korisnik koji nema bar jedan program ove vrste u svojoj kolekciji. Ovde ćemo ukratko opisati samo dva ovakva programa iz cele gomile sličnih, a njihov izbor je bio uslovljen kako njihovim kvalitetom i mogućnostima, tako i subjektivnim odnosom i jednogodišnjim iskustvom autora ovog teksta u radu sa njima.

DISK DOCTOR

Ovo je jedan od najboljih i sasvim sigurno najkomforniji disk editor izuzetno velikih mogućnosti. Prvo što upada u oči pri radu sa ovim programom je izuzetno pregledan prikaz celog sektora i to na dva načina: HEX ili ASCII. U HEX prikazu svi bajtovi su predstavljeni heksadecimalnim brojevima, dok su u ASCII prikazu kodovi koji predstavljaju karaktere predstavljeni običnim crnim slovima, a svi ostali sivim (light) znacima u heksadecimalnom obliku. Tako je omogućen lak i komforan uvid u kompletan sadržaj sektora i istovremeno jednostavno razlikovanje tekstualnih poruka od ostalog sadržaja. Na raspolaganju su jednostavne opcije za pretraživanje, skok na određeno mesto i to u oba moda, a tu je i dalei u koji se može smestiti jedan sektor i kasnije vratiti ili upisati na neko drugo mesto. Pored toga, tu su i opcije za automatsku eliminaciju neispravnih sektora u pojedinačnom fajlu ili na celom disku, kao i „popravljanje" (repair) oštećenog diska ili fajla kao i vraćanje izbrisanih fajlova (ručno ili automatski). Izuzetno dobar je i HELP koji je podeljen po sekcijama i pored uputstava za rad sa programom sadrži i dosta podataka o strukturi diska uopšte. Iako da ga mnogi korisnici štampaju i koriste kao priručnik. Sve u svemu, radi se o vrlo uravnoteženom i upotrebljivom programu čija je jedina mana sporost pri popravljanju loših sektora na disku — kad se rešite da koristite ovu opciju možete slobodno da pustite program da radi i odete na kafu. Sve ostale opcije su vrlo upotrebljive i dobro izbalansirane.

DUST

Po komforu i preglednosti nešto slabiji od prethodnog, ali po mogućnostima jači, DUST je jedan od najkapacitetnijih disk editora. Pored editovanja samog diska i pojedinačnih fajlova, omogućeno je i editovanje proizvoljnih blokova memorije i određivanje blokova proizvoljne dužine, a veličina bafera nije ograničena samo na jedan sektor, već količina podataka koju može da primi zavisi samo od količine slobodne memorije. Sadržaju bafera se mogu dodavati novi blokovi, tako da je moguće kombinovanje više raznih blokova u jednu celinu, a pored toga kompletan sadržaj se može snimiti na disk u obliku zasebnog fajla. To je naročito u našim (još uvek pretežno piratskim) uslovima vrlo korisna opcija jer omogućava čuvanje kopija FAT-ova i direktorijuma za svaki važniji disk, čime se smanjuje moguća šteta koju mogu da naprave razni virusi i druga štetočine koja se lako prenose i razmnožavaju u postojećim, slikovito rečeno nehigijenskim uslovima slobodnog kopiranja.

Na kraju treba reći da i ovaj program poseduje vrlo dobar HELP koji se takođe može odštampati, ovde čak i iz samog programa i kao i prethodni program nudi celovitiju informaciju o strukturi diska.

HELP programi

U tesnoj vezi sa „pravim" disk editorima je i jedna grupa manjih programa specijalizovanih isključivo za traženje i popravak loših sektora. Ovi programi se ne mogu nazvati nikakvim zajedničkim imenom izuzev onoga koje je dato u naslovu i koje jedino odvojeno slikovito opisuje njihovu namenu, a uz to se većini takvih programa najčešće i daju baš takva imena. Jedino što ovi programi mogu da ponude je prilično efikasno pronalaženje loših sektora na disketi praćeno opširnijim ili manje opširnim izveštajem, njihov u težim slučajevima prilično sumnjiv popravak i eventualno standardno formatiranje. Jedan od takvih programa, nazvan baš **HELP PRG**, nudi i opciju za reformatiranje uz očuvanje kompletnog sadržaja, što u nekim slučajevima može da bude vrlo, vrlo korisno. Pored toga, ovakvi programi su po pravilu male dužine i ne rade pod GEM-om, što znači da je njihovo učitavanje vrlo brzo, količina memorije koju zauzmu vrlo mala i najčešće ne morate da grickate nokte čekajući na rezultate.

Sve u svemu, nije loše imati neki od ovakvih programa, ali se ne isplati trošiti mnogo vremena i novca na njihovu nabavku.

Formatiranje i kopiranje

Kopiranje i formatiranje, mada dva suštinski različita posla, su uvek u bliskoj vezi, naročito u našim uslovima kada je poželjno smestiti što više podataka na što manje disketa. U principu svi bolji programi za kopiranje sadrže i opcije za formatiranje i to zajedno sa kopiranjem pri čemu se jednostavno kopira format izvornog diska ili odvojeno. Pri odvojenom formatiranju obično se nudi bar dva-tri različita formata: jedan standardni (80 traka, 9 sektora) i par proširenih formata od kojih su najčešći oni koji daju 82 trake sa po 10 sektora.

Odvojeni programi za formatiranje su isključivo namenjeni postizanju proširenih formata i omogućavaju postavljanje celog niza parametara bitnih za formatiranje. Mada mnogi korisnici koriste omiljene kopi programe i za formatiranje, autor ovog teksta je mišljenja da je bolje koristiti posebne programe za formatiranje. Pri tom posebno treba istaći da je sigurnost podataka na disketi utoliko sigurnija ukoliko je format bliži standardnom formatu i taj format bi u svakom slučaju trebalo koristiti za sva važnije programe i podatke, bez obzira na eventualnu uštedu pri korišćenju većih formata. Na vama je da procenite da li više vredi dva-tri diskete

ili čak i najmanja mogućnost da ostanete bez rezultata višenedeljnog ili vrlo verovatno i višemesečnog rada. Ako već koristite prošireni format tada obavezno koristite kvalitetne diskete koje je proizvođač deklarisao za takav format (npr. MAXELL RD koje su sertifikovane na 1 MB) i maksimalni format koji vaš drajv podržava, a pored toga obezbedite sigurnosne kopije (back-up) na standardno formatiranim disketama.

HYPERFORMATTER

Ovo je veoma dobar i pouzdan program koji je pored toga i *public domain*, a o njegovom kvalitetu odvojeno govori i činjenica da je izašlo već nekoliko verzija (jedna od novijih je 2.22). Program ne radi pod GEM-om, ali to ipak ne smeta mnogo. I bogato je naoknađeno mogućnostima koje pruža i brzinom učitavanja. Opcije su raspoređene u dve grupe, jedna ispod druge, i predstavljene su punim nazivima, s tim što se za selektovanje svake od njih koristi samo prvo slovo. Omogućeno je posebno biranje broja traka (maksimalno 86!), broja sektora po traci (do 11), interiv, verifikacija i broj foldera, a omogućeno je i odlučivanje o tome da li će disketa biti izvršna ili ne. Pored toga, omogućeno je i menjanje but sektora, a dat je i mini-HELP ali na nemačkom (do sada ste sigurno već naslutili da je reč o programu iz Nemačke škole). Uz program se dobijaju još dva mala programa, od kojih jedan služi za merenje brzine okretanja diska, a drugi za ubrzano učitavanje i pisanje. Merenje brzine okretanja diska, mada na prvi pogled čisto informativnog značaja, može da pruži odgovor na pitanja tipa „zašto ovaj disk ne prepoznaje ovaj format?". Program za ubrzavanje diska mada radi ono što treba, ne bismo mogli mirna srca da preporučimo, jer cena koja se plaća za igranje hardverom ume ponekad da bude neprijatno visoka.

COPYSTAR

CopySTar se najkraće može opisati kao solidan i upotrebljiv kopi program koji pored svih potrebnih opcija za kopiranje sadrži još i opcije za formatiranje. Kopiranje se može vršiti u dva moda — *normalnom* i *zaštićenom*, i uz to izabrati jednostruko ili višestruko kopiranje (više kopija istog uzorka), sa istovremenim formatiranjem (preslikava se format izvorne diskete) ili bez, sa verifikacijom ili bez nje (to je tzv. *ubrzano kopiranje* u koje se baš ne treba puno pouzdavati). Osnovna karakteristika zaštićenog kopiranja je kopiranje i diskova sa sistemskim i skrivenim fajlovima, dok normalno kopiranje vrlo često uspeva da iskopira čak i ono što većina ostalih programa kopira samo u zaštićenom režimu.

Opcija za formatiranje nudi tri formata (80 traka sa po 9 sektora, 81/10 i 82/11) i nije baš prebogata, ali ipak nudi sasvim dovoljno za prosečne potrebe i obezbeđuje kontinuitet rada i zaboravnim korisnicima koji ustanove da nemaju pri ruci formatiranu disketu tek kad već učitaju program. Posebna poslastica verzije 2.2 (ili većih, ako ih ima — ova je poslednja koju je video autor ovog teksta) je prikazivanje toka formatiranja i obeležavanje loših sektora kao i broja pokušaja za svaki sektor pri uključenoj verifikaciji. Ovo omogućava direktan uvid u kvalitet svake pojedinačne diskete, što se može pokazati kao dragoceno uvek kada na takvu disketu treba smestiti neki važan program ili fajl.

ST COPY

Ovaj progam se često može naći i pod imenom **ACOPY**, što se tumači na dva načina — jedno tumačenje je da je to skraćenica od **AMIGA COPY** što nema mnogo logike, a drugo, mnogo verovatnije tumačenje je da mnogi korisnici preimenuju najčešće korišćene programe tako da počinju slovom A. Na taj način se pri standardnom sortiranju po imenu takvi programi nađu pri vrhu prozora. Pošto su u našim uslovima pirati najčešći izvor svih, a naročito kopi programa, i znajući njihov način rada, ovakvo tumačenje je mnogo verovatnije.

Ovo je po opcijama za kopiranje sasvim prosečan kopi program, ali je po komforu i opcijama za formatiranje ubedljivo najbolji, tako da ga mnogi korisnici koriste pre svega za samo formatiranje. Izbor mogućih formata je zaista veliki, a omogućeno je posebno biranje broja traka (maksimalno 85) i sektora (maksimalno 11), a na raspolaganju je i opcija za dobijanje detaljnih informacija o formatu diskete.

PROCOPY

Poslednja (poznata) verzija ovog inače izuzetno nekomfornog ali vrlo brzog i sposobnog programa je 4.01. Napisan je tako da maksimalno koristi raspoloživu memoriju tako da se ceo proces kopiranja na megabajtnim mašinama može obaviti čak i u komadu, što je vrlo značajno za one korisnike koji takve poslova obavljaju na veliko i u žurbi. U radu je dosta pouzdan, ali izuzev preslikavanja formata izvorne diskete ne nudi nikakve druge opcije za formatiranje i uopšte je prilično neljubazan prema korisniku.

Za sva kopi programa bi se mogla dati jedna opšta karakteristika: mogu da budu vrlo korisni, naročito za formatiranje i ubrzavanje rada, ali pošto svi od reda jednostavno preslikavaju kompletan sadržaj diska, neizbežno je i prenošenje raznog „đubreta", odnosno ostataka izbrisanih pa presnimljenih fajlova i još kojećega drugoga. Sve to nastaje uglavnom kao proizvod fragmentacije diska i taloži se vremenom, tako da je najbolji i najčistiji način kopiranja onaj pod deskom, odnosno sa prozora-na-prozor (podrazumeva se da je sortiranje izvršeno po veličini). Na taj način se prenose samo fajlovi bez ikakvog „đubreta" (naravno, bez upuštanja u ocenjivanje kvaliteta samih programa koji se prenose).

Programi za arhiviranje

Ovi programi se po svojoj nameni usko nadovezuju na prethodnu grupu, to jest štede prostor na disketama. Osnovni princip njihovog rada je najkraće i najgrublje rečeno komprimovanje podataka tako što se svi srodni bajtovi sabijaju i podaci o tome zapisuju na disk. Postoji nekoliko metoda takvog komprimovanja, a za sva je karakteristično to da se najbolji rezultati postižu sa tekstualnim fajlovima i slikama (45—55%), a najslabiji sa samim programima (oko 30%). Na taj način komprimovani programi i fajlovi se ne mogu direktno koristiti, već se pre toga moraju dearhivirati ili „raspakovati", tako da je ovaj način štednje disketa pogodan samo za one korisnike koji imaju izuzetno korisnu naviku da prave sigurnosne kopije svih važnijih programa i fajlova. Pored toga postoje i programi koji instaliraju RAM disk koji radi na sličan način i to je prava stvar u svim slučajevima memorijske oskudice. Takav RAM disk se ponaša kao i svaki drugi, s tom razlikom što sve podatke „spakuje" pri smeštanju i „raspakuje" pri pozivanju. Cena koju za ovo treba platiti je izvesno usporenje čitanja i pisanja, ali pošto je reč o RAM disku to usporenje je taman toliko da se izjednači sa brzinom prosečnog hard diska. Tipičan primer takvog programa je **MAXIDISK.PRG**, koji se može startovati i sa diska i iz AUTO foldera, u kom slučaju mu se može dodati i mali .INF fajl i tada se postavlja automatski, pri čemu može i da kopira naznačene podatke sa disketa.

ARC

Ovaj program je prenesen sa PC-a, gde je stekao svetsku slavu, a samo prebacivanje je išlo vrlo glatko budući da je rađen u C-u. U ARC-u su Vam na raspolaganju brojne mogućnosti, počev od arhiviranja kompletnog sadržaja diskete ili foldera. Kad je arhiva jednom formirana, moguće je ispisati njen sadržaj na dva načina (samo spisak sadržaja ili kompletan sadržaj sa svim podacima o realnoj dužini, vrsti itd.), dodavati nove fajlove, izvlačiti pojedinačne fajlove sa ili bez brisanja u samoj arhivi itd. sve do izvršavanja pojedinih arhiviranih programa. Formiranje arhive i sve ostale pojedinačne operacije se mogu izvesti na dva načina: prvo korišćenjem jednostavnog linijskog interpretera, odnosno aktiviranjem programa **ARC.TTP**, koji zatim daje dijalog u kome se zadaju linijske komande. Drugi način je korišćenje programa **ARCSHELL.PRG**, koji predstavlja grafičku školjku za **ARC.TTP**, odnosno omogućava izdavanje komandi preko GEM-komandi. U drugom slučaju na ekranu se pojavljuje niz polja sa opcijama čije aktiviranje se vrši mišem na standardan način. Naravno, ovaj program je duži i zahteva više memorije i vremena za učitavanje.

Anti-virus programi

Kao što i samo ime govori, ovo je grupa programa koji su namenjeni otkrivanju i uništavanju virusa na disketama, a neki od njih imaju i funkciju imunizacije. Zbog toga ih mnogi korisnici popularno nazivaju *vakcinama*, što je i sasvim odgovarajuća analogija. Za one korisnike koji još ne znaju tačno šta su u stvari kompjuterski virusi (još ima i takvih srećnika!), da kažemo reč-dve i o tome. Najkraće rečeno, virus je mali program koji se smešta u but-sektor diskete i pri svakom startovanju mašine (butovanju) se učitava u memoriju gde nalazi neko manje prometno mesto odakle može nesmetano i na svoju ruku da bdi po mašini. Pri tome promeni nekoliko sistemskih varijabli tako da sva komunikacija sa diskom ide preko njega, čime obezbeđuje sebi mogućnost da se upiše na svaku nezaštićenu disketu koja se stavi u drajv. To samo po sebi ne bi predstavljalo neku veliku smetnju jer su sami virusi veoma mali, kad pri tome ne bi radili razne druge nažalijene stvari. Tipična šteta koju jedan virus može da napravi je **formatiranje diska i upisivanje raznog đubreta**, iz čega logično proizilazi da su svi podaci koji su se na takvom disku nalazili uništeni ili im je bar onemogućen pristup, što se u krajnjem slučaju svodi na isto. Pri svemu tome najveću prednost daje hard diskovima na kojima je u stanju da napravi pravi haos i vrlo često i namerljivu štetu jer na taj način uništi ogromnu količinu podataka, ali ni šteta koju može da napravi na disketama nije zanemarljiva. Njihovo širenje po svom mehanizmu umnogome podseća na neke ljudske zaraze, i shodno tome za to su najpogodniji upravo uslovi nekontrolisane razmene softvera kakvi vladaju u našim uslovima. Da bi se ovakva jednom „zakačena" štetočina na vreme otkrila i eliminisala, potrebno je mnogo vremena, znanja i živaca, pa su zbog toga napisani specijalizovani programi. Jedan od takvih programa je **VDU 2.4**, koji može da prepozna nekoliko virusa i veliki broj izvršnih but sektora. Pri tome upozorava korisnika da je pronašao virus ili da je reč o neimunizovanoj disketi koja nije imunizovana. Korisnik tada može da odluči da li da se otkriveni virus uništi i disketa pri tome imunizuje, odnosno da li da se imunizacija provede na čistoj ali neimunizovanoj disketi. Program je takođe *public domain* i uključen je u našu biblioteku.

Ei — „Štampana kola", Niš

Ljudi i multilejeri

Od dana kada se pre nešto više od 35 godina pojavila kao sastavni deo elektronskih sklopova, štampana ploča i kao proizvod i kao tehnologija sve do danas doživljava izuzetno dinamičan razvoj. Fabrika štampanih kola iz Niša, koja u nekoliko nastavaka gostuje na stranicama „Računara", spada u ovoj oblasti u red najnaprednijih proizvođača u našoj zemlji.

Prvobitna štampana ploča koja je imala provodnike samo sa jedne strane i kakvu bi „danas sva deca uradila za ručni rad" bila je u SAD dugo vremena važan strateški proizvod a tehnologija izrade — strogo čuvana vojna tajna. Otkako je pre 35 godina iz vojne prešla u civilnu industriju — da bi dala impuls razvoju američke privrede — štampana ploča je postala je jedna od najvažnijih komponenti savremene elektronike i jedan od pokretača njegovog burnog razvoja.

Proizvodnja štampanih ploča u Jugoslaviji je počela pre oko 25 godina, znači desetak godina za razvijenim zemljama, i ta se distanca uglavnom zadržala do danas. Ništa neobično. Nosioci razvoja štampanih ploča su razvijene zemlje jer je on uslovljen drugim visokim tehnologijama, a značajna ulaganja u kadrove, istraživanja i razvoj su od prvorazrednog značaja.

Štampane ploče koje se danas proizvode mogu se svrstati u nekoliko osnovnih grupa: jednoslojne ploče, dvoslojne sa metalizovanim rupama, fleksibilne ploče i multilejeri, odnosno višeslojne štampane ploče. Proizvodnja ovih poslednjih i za razvijene zemlje spada u visoke tehnologije, a pojedine tehnološke operacije i materijali u grupu brižljivo čuvanih privrednih tajni. Ovo naročito važi za multilejere sa preko 40 slojeva i malim prečnicima rupa (ispod 0,3 mm).

Danas u našoj zemlji postoji više desetina proizvođača štampanih ploča — malih i većih, privatnih i društvenih — ali samo mali broj spada u grupu naprednih proizvođača (naravno, naprednih za naše prilike). Kod takvih proizvođača tehnologija je stigla do klasičnog multilejera i to sa ograničenim brojem slojeva (negde oko 8). Složenije forme multilejera — kao što su čvrsto-fleksibilni, zatim multilejeri sa unutrašnjim i slepim prespojnim rupama, dvostruki multilejeri itd. sa popunjenim rupama — i po potrebama i po mogućnosti spadaju u bližu ili dalju perspektivu.

Značajan doprinos razvoju štampane tehnika u zemlji, a samim tim i razvoju elektronike, data je fabrika štampanih ploča Elektronske industrije u Nišu — „Ei — Štampana kola". Mada proizvodnja štampanih ploča u Ei-u postoji — u raznim organizacionim oblicima — oko 25 godina, fabrika „Štampana kola" kao specijalizovani proizvođač štampanih ploča je novijeg datuma i ove godine navršava 10 godina samostalnog postojanja.

Za to vreme, fabrika je značajnim ulaganjima u savremenu opremu, tehnologije i kadrove imala intenzivan razvoj, tako da se po obimu proizvodnje, asortimanu, nivou kvaliteta proizvoda i uopšte dostignutom tehnološkom nivou danas može svrstati u red vodećih i tehnološki najnaprednijih proizvođača štampanih ploča u zemlji.

Proizvodni program fabrike „Štampana kola" čine:

— jednoslojne štampane ploče, klasične i za SMD — tehnologiju

— dvoslojne ploče sa metalizovanim rupama čiji kvalitet zadovoljava domaće i mnoge strane tehničke uslove i



— višeslojne štampane ploče — multilejeri do 24 sloja.

Multilejeri su najsloženiji oblik štampanih ploča koje se rade u fabrici „Štampana kola", pri čemu se mora znati da su neke složenije forme multilejera (koje se kod nas još ne rade) trenutni tehnološki domet u proizvodnji štampanih ploča uopšte.

Savremeni elektronski uređaji, računari i drugi sistemi iz dana u dan postaju sve složeniji u pogledu performansi, a istovremeno su sve manjih dimenzija. Njihova proizvodnja se uglavnom zasniva na primeni multilejera, jer oni istovremeno omogućavaju najveću gustinu pakovanja elemenata — klasičnih i SDM — uz visok stepen pouzdanosti.

Osvajanjem proizvodnje jednog kompleksnog proizvoda kakav je multilejer, fabrika „Štampana kola" je ovladala svim pratećim tehnologijama kao što su:

1 — Priprema kompjuterskih podataka za izradu multilejera

2 — Mehanički i mokri proizvodni procesi (CNC — bušenje i obrada gabarita, foto-postupci, galvanski i drugi procesi)

3 — Testiranje proizvoda i kontrola procesa

Priprema kompjuterskih podataka

Dve glavne faze prethode izradi multilejera.

Prva faza je projektovanje, u kojoj se ideja projektanta prevodi, pre svega korišćenjem CAD sistema, u različite oblike podataka. Projektovanje obuhvata:

— unošenje u sistem podataka o konkretnom dizajnu

— proveru funkcionalnosti i ispravnosti dizajna

— korekcije i optimizaciju

— generisanje izlaznih baza podataka

Druga faza je priprema različitih oblika proizvodne dokumentacije i ovaj deo se radi u fabrici štampanih ploča.

Polazni materijal su podaci koji su generisani na kompjuterskim radnim stanicama na kojima je rađeno projektovanje, u vidu datoteka na podesnim medijima (najčešće flopy disketama) i crteži koji dopunjuju potrebne podatke za proizvodnju.

Obradom polaznih podataka dobija se proizvodna dokumentacija koja uglavnom sadrži filmove signalnih slojeva, filmove za VCC i GND, solder masku i slično, i programe za CNC — bušenje i obradu gabarita ploča.

Procesne tehnologije

Proizvodnja multilejera je veoma složen i skup proces i sastoji se od nekoliko desetina tehnoloških operacija, od kojih je svaka kritična. Samo vrhunske tehnologije mogu da obezbede potreban kvalitet proizvoda, a da to ima i odgovarajuću ekonomsku opravdanost.

Fabrika „Ei — Štampana kola" je uspešno ovladala svim tim tehnologijama, među kojima su najznačajnije:

— Tehnika pasovanja odnosno slaganja slojeva koje omogućuje izradu multilejera sa preko 20 slojeva

— Bušenje otvora kompjuterski kontrolisanim bušilicama koje omogućuju visoku tačnost pozicioniranja i daju potreban kvalitet izbušenih otvora

— Priprema rupa za metalizaciju, metalizacija i galvansko nanošenje lemne legure (Sn Pb 60 40) sa infra crvenom prekristalizacijom lemne legure

— Pozlata konektorskih izvoda sa podslojem nikla

— Suvi foto-postupak kojim se mogu izvući i provodne linije širine ispod 0,2mm

— Obrada gabarita ploča kompjuterski kontrolisanim mašinama

Tehnološki postupak izrade multilejera sa 4, 6 i 8 slojeva atestiran je za vojnu proizvodnju, s tim što se atest za 8-slojnu ploču može proširiti i na ploče sa više od 8 slojeva.

Kontrola procesa i testiranje proizvoda

Samo strogo kontrolisan proces u svim proizvodnim fazama može da obezbedi potreban kvalitet štampanih ploča. Proces proizvodnje multilejera kontroliše savremeno opremljena laboratorija metodom mikro-preseka. Svakodnevno se kontrolišu svi bitni parametri koji određuju kvalitet multilejera:

— debljina bakarne prevlake u rupama i na provodnicima,

— debljina i kvalitet ostalih prevlaka (zlato, nikal, kalaj-olovo)

— dubina i kvalitet „nagrizanje unazad"

— poroznost prevlaka i slično

Električna ispravnost gotovih ploča kontroliše se na automatskom testeru koji, prema zadatim podacima, lako pronalazi i odbacuje sve ploče sa prekidima veze, kratkim spojevima i drugim električnim nedostacima, što korisnicima štampanih ploča garantuje apsolutnu sigurnost u proizvodnji.

Ovakav pristup proizvodnji štampanih ploča učinio je da se „Ei — Štampana kola" može svrstati u red vodećih proizvođača u zemlji. Pri svemu ovome ne sme se zanemariti jedna značajna istina: i pored izuzetne kompleksnosti proizvodnog procesa, multilejere ipak prave ljudi, a ne oprema. Čini se da je u tome i najveća snaga „Štampanih kola".

Jovan Protić

U sledecem broju

U sledećem broju ćemo pisati detaljnije o projektovanju štampanih ploča pomoću računara, savremenim tehnologijama proizvodnje štampanih ploča i kontroli i testiranju kvaliteta.

POSTAVLJANJE STANJA PALETE
Code: 5—5D Set Palette Driver State

```
VOID vsp_state (handle, port, film_num,
                lightness, interlace,
                planes, indexes)
     WORD handle
     WORD port
     WORD film_num
     WORD lightness,
     WORD interlace
     WORD planes
     WORD indexes [8] [2],
```

Svi parametri imaju isto značenje kao i u prethodnoj funkciji s tom razlikom što su ovde to vrednosti koje se postavljaju a ne vrednosti koje se ispituju.

SPREMANJE STANJA PALETE
Code: 5—5E Save Palette Driver State

```
VOID vsp_save (handle)
     WORD handle
```

Sprema sve parametre stanja palete na disk.

ONEMOGUĆAVANJE PORUKA
Code: 5—5F Suppress Palette Messages

```
VOID vsp_message (handle)
     WORD handle
```

Ovom funkcijom se onemogućava ispisivanje poruka na ekranu od strane drajvera uređaja za snimanje filma.

ISPITIVANJE GREŠKE
Code: 5—60 Palette Error Inquire

```
WORD vqp_error (handle)
     WORD handle
```

Vraća kod greške ili poruke koju drajver snimača želi uputiti korisniku. Ova funkcija ne uništava signal greške i on će se pojaviti na ekranu ukoliko funkcija **vsp_message** nije pozvana. Postoji jedanaest kodova greški:

0 — nema greške
1 — otvorena je komora za film
2 — ne postoji port na zadatoj lokaciji
3 — paleta nije pronađena
4 — video kabl nije spojen
5 — OS ne dozvoljava alociranje memorije
6 — nema dovoljno memorije za bafer
7 — memorija nije dealocirana
8 — drajverski fajl nije pronađen
9 — drajverski fajl nije korektan
10 — poruka korisniku da obradi film

Prelazi za Metafile uređaj
(Metafile Escapes)

Sa Metafile uređajem ste se već sreli pri samom otvaranju radnih stanica. Ovaj je pseudo-uređaj potencijalno najmoćnije oružje GEM-a. Ono što njemu pošaljete tj. snimite na disk, svaka mašina sa GEM-om (dakle ST i PC) će s lakoćom učitati i iskoristiti. Isto važi i za razne programe za crtanje te za DTP (DeskTop Publishing) programe. Nevolja je međutim u tome što postoje samo tri dokumentovane funkcije za rad sa ovim uređajem i sve tri su namenjene upisivanju. Postojanje brojnih programa koji intenzivno koriste Metafile (što prepoznajete po njegovim produktima — fajlovima sa nastavkom .GEM) govori međutim da li postoje funkcije za koje samo proizvođači softvera znaju ili je neko iscrpno analizirao strukturu ovog fajla pa sa njim radi kao sa običnim skupom bajtova.

AŽURIRANJE OKVIRA
Code: 5—62 Update Metafile Extents

```
VOID v_meta_extents (handle, min_x,
                     min_y, max_x, max_y)
     WORD handle
     WORD min_x, min_y
     WORD max_x, max_y
```

Ovom funkcijom se podešavaju koordinate gornjeg levog i donjeg desnog temena pravougaonika u kome se nalaze sve stavke upisane u Metafile uređaj. Pravougaonik dakle služi kao okvir i po njemu se ravna program koji podatke učitava.

UPISIVANJE STAVKE
Code: 5—63 Write Metafile Item

```
VOID v_write_meta (handle, num_intin,
                   num_ptsin, intin,
                   ptsin)
     WORD handle
     WORD num_intin, num_ptsin
     WORD intin [ ], ptsin [ ]
```

Upisivanje stavke se svodi na upisivanje delova **intin** i **ptsin** nizova za koje je broj aktivnih elemenata zadat parametrima **num_intin** i **num_ptsin**. Element **intin 0** mora sadržati kod **0x65** ili veći čime se označava da se radi o korisnički definisanoj stavki. Podatak da se na isti način upisuju i sistemske stavke (izdavanjem geometrijskog lika na radnoj stanici za Metafile uređaj) može pomoći svima onima koji žele analizirati GEM fajlove.

PROMENA IMENA
Code: 5—64 Change GEM VDI Filename

```
VOID vm_filename (handle, filename)
     WORD handle
     CHAR filename [8]
```

Fajl u koji se snimaju kodovi komandi predati Metafile uređaju ima inicijalno ime **GEMFILE.GEM**. Ovom se funkcijom to ime može zameniti fajlom proizvoljnim imenom zadatim parametrom **filename**.

Žarko Berberski

GEM za programere

prvi deo: V D I

GEM je skraćenica od Graphics Environment Manager (upravljač grafičke okoline) i sloga se čita baš tako kako je i napisano — „gem". Neki američki programeri imaju međutim običaj da ovu skraćenicu izgovaraju kao „džem" što nije naziv za gusti pekmez već reč koja u engleskom jeziku znači dragulj. Radno ime u toku razvoja ovog paketa bilo je naime Crystal (kristal) i odatle vuče koren čitava zbrka oko toga kako se ime izgovara i šta znači. GEM je dakle jednostavna skraćenica punog naziva za jedan grafički paket koji je uzgred daleko od toga da bude kristal ili dragulj. Radi se naime o sasvim prosečnom paketu koji olakšava život korisniku ali ga zagorčava programeru koji mora da izvrši čitav niz inicijalizacionih postupaka pre nego što uopšte i pomisli na to da počne da koristi funkcije ovog paketa.

U osnovi je čitav paket zamišljen kao kompletan operativni sistem koji bi s jedne strane imao skup funkcija vrlo niskog nivoa za direktno upravljanje hardverom a s druge strane bi bile funkcije koje bi radile samo sa par apstraktnih tipova objekata i omogućavale kako jednostavno građenje grafičke korisničke okoline tako i višeprogramski rad (tj. raspoređivanje procesorskog vremena na više programa za koje bi korisnik mislio da rade istovremeno). U startu se međutim moralo odustati od prauzimanja svih funkcija operativnog sistema. Operativni sistemi postoje naime suviše dugo i imaju suviše mnogo korisnika i već napisanih aplikativnih programa da bi se mogli lako baciti i sve početi izpočetka. Tako je GEM praktično sveden na paket koji brine o grafičkoj okolini i delimično o višeprogramskom radu, a sve poslove oko rada sa diskom, raspodelom memorije itd. prepušta standardnom operativnom sistemu.

U raznim polovičnim objašnjenjima koje se nalaze po stranoj literaturi pojavljuje se više načina deljenja funkcija GEM-a, zavisno od toga da li je autor više naklonjen mašlanju o tome kako bi stvar trebalo izgledati ili od toga kako ona izgleda danas. Pošto ovaj tekst treba pre svega da vam posluži kao minimalna dokumentacija za korišćenje danas postojećeg sistema, takva će biti i podela.

GEM se dakle sastoji iz **dva glavna dela**. Prvi se naziva **VDI (Virtual Devise Interface)** i obezbeđuje osnovne funkcije za iscrtavanje jednostavnih grafičkih oblika na raznim uređajima kao i za unošenje grafički orjentisanih podataka sa raznih uređaja. Drugi deo se naziva **AES (Application Environment System)** i njegova je uloga da brine o radu sa složenim i strukturiranim grafičkim objektima (prozorima, dijalozima, ikonama, menijima itd.) kao da obezbeđuje raspodelu procesorskog vremena na više programa koji se korisniku prikazuju kao programi koji rade paralelno.

VDI se sa svoje strane sastoji iz dva osnovna dela. Prvi je zadužen za učitavanje i instaliranje programa za pogon perifernih uređaja koji se obično nazivaju **drajverima** (što je transkripcija engleske reči driver u značenju vozač). VDI nam uvodi pojam opšte radne stanice koja ima svoju radnu površinu po kojoj se iscrtavaju grafički oblici ili sa koje se uzimaju grafičke informacije (položaj tačke, odabir nekog posebnog tastera, vrednost neke fizičke veličine itd.). Svakom uređaju se dodeljuje jedna fizička radna stanica čijim se otvaranjem u stvari učitava i instalira drajver koji omogućava komunikaciju s konkretnim uređajem. Za jednu fizičku radnu stanicu se može otvoriti više virtualnih tako da jednom uređaju može pristupati više programa koristeći jedan isti drajver. Najniži deo GEM-a se ne nalazi u ROM-u, već se pri podizanju mašine učitava sa diska, instalira i ostaje razdeljen baš kao da je bio u ROM-u. Informacije o svim raspoloživim drajverima uzima iz ASSIGN.SYS fajla, no same drajvere učitava tek kad neki program to zatraži. Ukoliko koristite samo ekran (što u VDI-terminologiji podrazumeva displej, tastaturu i miša — ono što se u TOS-u naziva konzolom) nema potrebe za učitavanjem GDOS programa (što je službeno ime ovog dela VDI-a i naziv fajla na disku) jer odgovarajući drajveri za njega već postoje u ROM-u. Sve to uz uslov da ne želite da koristite više fontova, jer tada je GDOS neizbežan.

Drugi deo **VDI-a** predstavlja biblioteku funkcija koje omogućavaju korisniku relativno komforan rad sa osnovnim grafičkim oblicima, kao i unošenje i izdavanje podataka preko perifernih uređaja (i ekran se smatra perifernim

uređajem). Sastoji se od sedam kategorija: **kontrolne funkcije, funkcije izdavanja, atributska funkcije, rasterske funkcije, funkcije unosa, funkcije upita i funkcije prelaza** — sve u svemu preko 110 funkcija. Ako ste pomislili da paket od 110 funkcija mora da sadrži sve ono što programer ikada može poželeti — prevarili ste se. Dok su jedne funkcije previše usitnjene, druge su potpuno ispuštene. Drastičan primer je spremanje objektno zadate slike (tj. slike koja nije skup tačaka već skup kodova za crtanje osnovnih geometrijskih oblika VDI funkcijama): gde postoje funkcije za spremanje u fajl na disku, ali ne i ona za učitavanje iz njega. Uprkos tome, brojni grafički programi uspevaju to da urade. Kako — za sad je prekriveno velom tajne.

Drugi deo **GEM-a, AES** sa oko 65 funkcija, podeljenih u 11 biblioteka zadužen je za kreiranje, promenu, održavanje i uništavanje složenih grafičkih objekata od kojih su najpoznatiji meniji, dijalozi, alerti (tj. upozorenja), prozori i ikone. Moguće je, međutim, napraviti proizvoljan objekt kombinujući elemente iz zadatog skupa. Takva proizvoljne složena objekta neki autori nazivaju slobodnim objektima a drugi panelima. Obično se sve definicije objekata smeštaju u poseban pomoćni fajl (prepoznaćete ga po nastavku RSC) čiji se sadržaj stvara i menja posebnim editorskim programima koji direktno prikazuju kako objekti izgleda na ekranu.

Drugo zaduženje **AES-a** je briga o raspodeljivanju vremena među raznim programima. Ta raspodela se odvija preko složenog sistema prekida. Kadgod program želi da unese neki podatak, on poziva neku od AES funkcije za rad sa događajima. Tog trenutka program gubi kontrolu (kaže se da ide u hibernaciju), a dobija je sledeći program iz posebne liste čekanja. Kad se dogodi ono što je program tražio (pritisak tastera, pomeranje miša itd.) kontrola mu se vraća, a onaj drugi program ide u hibernaciju. Na taj način se efikasno raspoređuje vreme među programima, pa korisnik ima utisak da ih je nekoliko aktivno istovremeno. Programi mogu međusobno da komuniciraju porukama koje opet putuju po sistemu posredstvom AES-a. Čitav prekidni sistem uglavnom dobro radi sa sadašnjim operativnim sistemima. Valjko je pitanje šta bi se desilo kad bi GEM morao da radi pod operativnim sistemom koji već ima rešeno raspoređivanje vremena i rad sa više programa.

Pošto AES, za razliku od VDI-a, nije samo skup raznih kategorija funkcija već i kompletan upravljački program sa neostvarenim željama da postane kompletan operativni sistem, to se i funkcije njegovih biblioteka ne mogu shvatiti kao obične funkcije već više kao ulazi u **upravljačke module**, koji se kratko nazivaju **upravljačima** i **menadžerima**. Svaka biblioteka odgovara jednom upravljačkom modulu, pa će od sada sve biti razmatrano u terminima tih modula, odnosno njihovih globalnih funkcija.

Postoji dakle 11 upravljačkih modula koji vrše sledeće funkcije: **upravljanje aplikacijama, događajima, menijima, objektima, oblicima, grafikom, „parčićima" podataka, izborom fajla, prozorima, pomoćnim (RSC) fajlovima i ljuskama**.

Rad AES-a je skoro kompletno dokumentovan, no neki programeri su ipak pronalazili kodove poruka koje i AES i većina programa prima, a nigde u zvaničnim dokumentima GEM-a nisu navedene. Za najveći broj programerskih potreba sasvim su dovoljna i one koja su dobro dokumentovane.

Radne stanice

Sistem radnih stanica kao softverskog modela nije izmišljen zbog GEM-a već mnogo ranije, sa pojavom prvih eksperimentalnih sistema sa interaktivnom grafikom. Može se reći da je ova koncepcija unificiranja rada sa svim grafičkim ulaznim i izlaznim uređajima kamen temeljac brzog i uspešnog razvoja grafičkih radnih okolina.

Ideja je u suštini sasvim jednostavna, ali i potpuno strana korisnicima kućnih računara koji su godinama navikavani da razmišljaju u terminima potpuno razdvojenih uređaja: displeja, tastatura, štampača itd. Pišući programe na mašinama koje jedva da raspolažu nečim što liči na operativni sistem, ogroman broj korisnika je nlazio sve dublje u zamku nepotrebnih hardverski zavisnih detalja. Slična stvar se događala i sa programerima koji su na većim mašinama radili grafičke programe. Svako je imao neku svoju ideju o tome kako komunikacija s korisnikom treba da izgleda, koje karakteristike štampača treba koristiti, kako treba upotrebljavati kursorske tastere itd. Sve je to vodilo gotovo nikakvoj prenosivosti programa i potpunom odsustvu komunikacije među ljudima koji su radili grafičke programa. Jedini način da se to preseče jednom za sva vremena bilo je uvođenje neke vrste standardizovanog jezika za rad sa grafikom i potpuno odvajanje hardverski zavisnog dela od programera koji pišu grafička programe. Pri tome je maksimalno korišćena ideja na kojoj se bazira pristup disku u praktično svim standardnim operativnim sistemima.

Kada želite da upišete neke podatke na disk ili da ih pročitate s njega, vi činite isto ono što i svaki kancelarijski službenik kad mu trebaju papiri s podacima — otvarate fajl odnosno fascikl u kojoj su podaci. Pri otvaranju fajla sistem mu dodeljuje neki broj po kome će uvek znati čemu tačno želite da pristupite. To što se otvaranje sastoji od izvesnog broja pomeranja glava diska i slanja i primanja neke količina vama nejasnih podataka uopšte vas ne uzbuđuje — vi ste otvorili svoj fajl, a za ostalo neka se sistem brine sam (zato ste ga i kupili).

Primenite sada istu tu ideju na rad sa štampačem. Za što bi vas bilo briga kako će biti povučena linija na štampaču ili nacrtan krug. Jednostavno „otvorite" štampač i pošaljite odgovarajuću komandu, a ostalo je briga sistema. Isti princip se može proširiti na displej, crtač, pa i mnogo egzotičnije uređaje kao što je kamera.

Kad želite da radite s nekim ulaznim ili izlaznim grafičkim uređajem, jednostavno mu **otvorite fizičku radnu stanicu** i dobijete **referentni broj**, među programerima poznat kao **handle**. Otvaranje fizička radna stanica podrazumeva učitavanje programa koji će direktno „terati" hardver (pomerati papir i glavu štampača, pero crtača, ispisivati tačke na ekranu itd.), među programerima poznatijeg kao **drajver**. Sve to, međutim, uopšte ne treba da vas interesuje. Vi ste otvorili svoju radnu stanicu i šaljete joj komande, a kako će drajver „ubediti" konkretan uređaj da urad to što treba — njegova je stvar.

```
WORD handle
CHAR filename [ ]
WORD aspect
WORD scaling
WORD num_pts
WORD xyarray[ ]
```

Ova funkcija učitava fajl sa bit-mapiranom slikom u standardnom GEM formatu (sa nastavkom .IMG) i izdaje je na zadatu radnu stanicu. Ime fajla se zadaje parametrom **filename**. Parametar **aspect** određuje odnos učitane i izdate slike. Ukoliko mu je vrednost 0, nikakvo skaliranje se neće izvršiti. Vrednost 1 označava da će se slika skalirati tako da se poštuju odnosi u učitanoj slici tj. da krugovi daju ostanu krugovi itd., dok vrednost 2 zahteva poštovanje formata stranice na koju se izdaje slika pa će tako za kompletnu sliku biti izdata potpuno popunjena stranica ta čaka ali će se originalni odnosi izgubiti. Parametar **scaling** može imati vrednost **0** za uniformno skaliranje (odgovara **aspect** = 1) ili 1 za različito skaliranje po **x** i **y** osi (odgovara **aspect** = 2). Nigde nije dokumentovan odnos parametara **aspect** i **scaling** tj. nije navedeno koji od njih ima viši prioritet odlučivanja. Ostaje u stvari nejasno zašto je uopšte uveden parametar **scaling** kad već **aspect** određuje sva karakteristike skaliranja.

Parametar **num_pts** određuje da li će se za određivanje izraza slike (koji će biti izdan na zadatoj radnoj stanici) biti korišćena oba temena učitane slike (0): gornje levo teme iz **xyarray** [] a donje desno iz učitane slike (1) ili oba temena iz **xyarray**[]. Niz **xyarray[4]** daje koordinate gornjeg levog i donjeg desnog temena pravougaonika koji se koristi za nalaženje izraza učitane slike koji će se izdavati na zadatom uređaju (obično je to ekran ili štampač). Korišćenje ovih temena zavisi od vrednosti parametra **num_pts**. Koordinate temena su zadate na standardni način:

(x1, y1) (xyarray [0], xyarray [1])

(x2, y2) – (xyarray [2], xyarray [3])

Uređajno-zavisni prelazi

(Device Specific Escapes)

IZBOR PALETE
Code: 5—3C Select Palete

```
VOID vs_palette (handle, palette)
     WORD handle
     WORD palette
```

Izbor jedne od dve palete za IBM kompatibilni kolor monitor. Ako je **palete** 0, izabrana je paleta (**red, green, brown**) a ako je **palete** 1, izabrana je paleta (**cyan, magenta, white**).

Funkcije prelaza snimača filma

(Film Recorder Escapes)

Snimač filma koji se u GEM terminologiji naziva i kamerom je uređaj koji proizvodi rastersku sliku na ekrančiću veličine dijapozitiva ili male fotografije. Često je u taj uređaj već integrisan i foto aparat za 35 mm film ili za film polaroid-tipa. DR se očigledno potrudio da dobro opskrbi svoj paket funkcijama za pogon ovog uređaja, verovatno se nadajući da će GEM jednog dana biti korišćen i na radnim stanicama. Kako danas stvari stoje sva ove funkcije su praktično neupotrebne na personalnim računarima koji su uglavnom jedine mašine koje „tera" GEM.

TIP FILMA
Code: 5—5B Inquire Film Types

```
VOID vqp_films (handle, film_names)
     WORD handle
     CHAR film_names [5] [25]
```

Vraća niz stringova u kojima su nazivi tipova filmova koje kamera snimača može eksponirati. Po definiciji ovih stringova može biti samo pet a svaki može biti dugačak do 25 karaktera. Međutim, dokle god se radi u C-u, asembleru ili Modul 2, nema nikakvih problema da ovih stringova bude i više jer se uvek prenosi samo adresa početka.

ISPITIVANJE STANJA PALETE
Code: 5—5C Inquire Palette Driver State

```
VOID vqp_state (handle, port, film_num,
               lightness, interlace,
               planes, indexes)
     WORD handle
     WORD *port
     WORD *film_num
     WORD *lightness
     WORD *interlace
     WORD *planes
     WORD indexes [8] [2]
```

Parametar ***port** daje redni broj porta preko koga se vrši komuniciranje s uređajem za snimanje dok ***film_num** sadrži redni broj filma (od 0 do 4) koji se koristi. Parametar ***lightness** daje relativnu osvetljenost pri eksponiranju filma (0 za normalnu, 3 za dvostruko povećanu i -3 za dvostruko smanjenu). Ukoliko parametar ***interlace** nije 0 trebaće vam dvostruko više memorije za sliku jer se ona tada generiše „preplitanjem" dve slike. Niz **indexes [8] [2]** sadrži indekse za osam boja koje su reprezentovane na pomalo čudan način. Svakoj boji odgovara jedan par brojeva. Oba broja su u stvari ASCII kodovi nekih karaktera pa im se opšti oblik može napisati kao **(WORD) character** ili **(int) character**. Većina C kompajlera automatski vrši ovu konverziju tako da se može pisati i direktno:

indexes [7] [0] = 'A'

Prvi broj u paru je iz intervala ['A' ... 'H'] = [41 ... 48] a drugi iz intervala ['1' ... '9'] = [31 ... 39]. Ovakvim se parom u stvari bira jedna boja iz tabele boja koja postoji u drajveru za snimač filma.

IZDAVANJE TEKSTA
Code: 5—0CV Output Alpha Text

```
VOID v_curtext (handle string)
     WORD handle
     CHAR string [ ]
```

Izdaje tekst zadat parametrom **string** počev od tekuće pozicije alfa kursora. Ekran naravno mora biti u alfa načinu da bi ova funkcija radila.

IZBOR NAČINA ISPISIVANJA TEKSTA
Code: 5—0D, 0E Select Alpha Text Style

```
VOID v_rvon (handle)
     WORD handle
VOID v_rvoff (handle)
     WORD handle
```

Postoje dva načina ispisivanja alfa teksta „normalni" (odabira se **v_rvoff** funkcijom) i „inverzni" (odabira se **v_rvon**) funkcijom.

ADRESA ALFA KURSORA
Code: 5—0F Inquire Alpha Cursor Address

```
VOID vq_curaddress (handle row column)
     WORD handle
     WORD *row *column
```

U parametrima ***row** i ***column** vraća tekuću adresu alfa kursora.

Opšti ekranski prelazi

(General Screen Escapes)

STANJE GRAFIČKE TABLE
Code: 5—10 Inquire tablet Status

```
WORD vq_tabstatus (handle)
```

Naziv ove funkcije je vrlo neprecizan jer se ona ne odnosi samo na grafičku tablu već na bilo koji lokator tj. uređaj kojim se zadaje pozicija tačke pomoću koordinata. Funkcija vraća vrednost **0** ako nikakav lokator nije priključen a vrednost različitu od nule (obično je to 1 ili -1) ako je lokator priključen.

IZDAVANJE „ČVRSTE KOPIJE"
Code: 5—11 Hardcopy

```
VOID v_hardcopy (handle)
     WORD handle
```

Pod „čvrstom" kopijom se podrazumeva svaka kopija koja ne nestaje gašenjem računara. Najčešće se radi o listu papira ali to može biti i eksponirani film. Ovom funkcijom se ne radnu stanicu štampača (iza koje se pak mogu skrivati razni uređaji) šalje bit-mapirana slika se zadate radne stanice. Najčešće se ovom funkcijom izdaje slika sa ekrana na štampač.

POSTAVLJANJE I UKLANJANJE GRAFIČKOG KURSORA
Code: 5—12, 13 Place and Remove Graphic Cursor

```
VOID v_dspcur (handle x y)
     WORD handle
     WORD x y
VOID v_rmcur (handle)
     WORD handle
```

Funkcijom **v_dspcur**, postavlja se grafički kursor u tačku sa koordinatama **x, y** a funkcijom **v_rmcur** se isti kursor uklanja. Obe funkcije se obično koriste pri unosu podataka **probom** jer tada VDI neće sam da izvrši ovo postavljanje već to mora program učiniti ako je potrebno.

Funkcije prelaza štampača

(Printer Escapes)

POMERANJE STRANICE
Code: 5—14 Form Advance

```
VOID v_form_adv (handle)
     WORD handle
```

Forsira prelazak na sledeću stranicu na štampaču ili crtaču.

IZDAVANJE PROZORA
Code: 5—15 Output Window to Printer

```
VOID v_output_window (handle xyarray)
     WORD handle
     WORD xyarray [ ]
```

Sadržaj pravougaonika zadatog elementima niza **pxyarray 4** na standardni način izdaje se na printer kao bit mapirana slika.

BRISANJE BAFERA ŠTAMPAČA
Code: 5—16 Clear Printer Display List

```
VOID v_clear_disp_list (handle)
     WORD handle
```

IZDAVANJE SLIKE IZ FAJLA
Code: 5—17 Output Bit Image File

```
VOID v_bit_image (handle filename aspect
     scaling num_pts xyarray)
```


Od trenutka kad se otvori fizička radna stanica, uređaj za vas prestaje da postoji. To međutim postavlja dodatne zahteve pred grafički sistem, jer sve radne stanice moraju da budu sposobne da prihvate iste komande. Crtaču na primer uopšte nije problem da nacrta elipsu ili kvadrat, ali kako prosečan štampač naterati da to uradi? Na displeju se opet može jednostavno popuniti bilo koja kontura, ali kako to uraditi na crtaču koji poznaje samo linije „specijalnost"? Primera ima mnogo. Najvažniji posao pri projektovanju grafičkog sistema baziranog na modelu radnih stanica je izbor skupa osnovnih grafičkih operacija koje će svi uređaji moći da izvršavaju, a koje će opet biti dovoljne za najveći broj programa.

VDI raspolaže grupom od desel opštih grafičkih likova koje svaki uređaj može da nacrta. Grupa se naziva **GDP** (Generalized Device Primitives) i sadrži funkcije za crtanje krugova, elipsi, pravougaonika, kružnih i eliptičnih isečaka i poravnatog teksta. Osim toga postoji još pet osnovnih funkcija koje ne spadaju zvanično u ovu grupu, ali su takođe obezbeđena na svim uređajima i služe za crtanje izlomljenih linija, nizova razdvojenih tačaka, popunjenih poligona itd. Da bi se sačuvala primenljivost čitavog sistema i na neke buduće periferne uređaje, pri otvaranju fizičke radne stanice mogu se dobiti tačni podaci u raspoloživosti osnovnih grafičkih likova. To omogućava nekim uređajima koji baš nikako ne mogu da nacrtaju neki lik da to na vreme prijave sistemu.

Da bi se crtanje uspešno kontrolisalo, potrebno je raspolagati mogućnošću podešavanja raznih **atributa**. Njime se određuje debljina i oblik linije, boja kojom se crta i boja pozadine, načina na koji se popunjavaju konture i uzorak kojim se popunjavanje vrši, način ispisivanja i podešavanje teksta itd. Vrednosti svih ovih atributa čine zajedno nešto kao ličnu kartu radne stanice. Od njih bitno zavisi šta će se i kako videti na izlazu (displeju, papiru, filmu), pa uvek treba dobro paziti i planirati kako će se atributi podešavati. Greške u ovom podešavanju vide se tek kad je program gotov i kada ne izlazu dobijete gomilu nejasno raspoređenih linija i mrlja, a ne ono što ste zamislili.

Otvarajući fizičku radnu stanicu još nismo rešili sve probleme. Pošto se GEM manje ili više uspešno ponaša kao višeprogramski sistem, morate uvek imati na umu da će još nekom programu trebati ista fizička radna stanica. Da se podaci ne bi mešali i stvarali pravu konfuziju, uvode se i **virtuelne radne stanice** koje se otvaraju na već otvorenoj fizičkoj radnoj stanici. Sa programerskog stanovišta virtuelna radna stanica se ponaša kao i fizička. Na isti način se otvara, šalju joj se iste komande i ona ih na isti način izvršava. Jedina je razlika u tome što i drugi programi mogu otvarati svoje virtuelne radne stanice i nesmetano pristupati istom fizičkom uređaju bez mešanja podataka. U svakom trenutku je naime aktivan samo jedan program, mada se korisniku može učiniti da ih više rade paralelno što je posledica brzine procesora i raspoređivanje vremena. Taj aktivan program dakle pristupa uređaju preko svoje virtuelne radne stanice, zadaje svoje atribute (oblik, debljina, boja linije, način ispisivanja, uzorak popunjavanja itd.) i crta svoje grafičke likove. Potom drugi program postaje aktivan i on sad koristi svoju virtuelnu stanicu sa svojim atributima i crtežima. Mada oba programa prikazuju podatke istim uređajem (na primer ekranom) njihove radne okoline su ipak razdvojene. U nekoj naprednijoj budućoj verziji će svaka virtuelna radne stanica imati i svoje bafere za sliku, pa će time i fizički izlaz biti razdvojen, ali tada će možda sve to biti fizičke radne stanice. Za sada *svaki uređaj može imati samo jednu fizičku radnu stanicu* ali njoj može odgovarati više virtuelnih.

Svaka radna stanica može koristiti jedan od dva koordinatna sistema. Prvi se naziva normalizovanim (**NDC — Normalized Device Coordinates**) a drugi rasterskim (**RC — Raster Coordinates**). Normalizovani sistem je zamišljen kao univerzalni i hardverski nezavisan koordinatni sistem sa početkom u donjem levom uglu i rasponom koordinata **[0..32767]**. Ovaj sistem se međutim retko koristi za obično crtanje jer zahteva suviše preračunavanja pri generisanju slike, a to bitno usporava rad upravljačkog programa. Rasterski sistem koristi realne koordinate uređaja, pa je zato crtanje u njemu mnogo brže ali i teže prenosivo na druge uređaje.

Sistemom radnih stanica nije unificirano samo izdavanje slika već i unošenje informacija. Svakoj radnoj stanici su dodeljena četiri **logička uređaja unosa** za koje programer zna samo opšti princip rada i tip podataka koji se njime unose. Ne znači je dakle opet striktno razdvajanje onoga što je potrebno za pisanje programa koji koristi uređaj od onoga što je potrebno da bi se taj uređaj direktno kontrolisao.

Prvi logički uređaj unosa se naziva **lokator**. To je uređaj koji je sposoban da unese poziciju tačke na radnoj površini radne stanice. **Radna površina** je opet softverski model za fizički medijum na kome se vrši ispis (korisni deo displeja, hartija u štampaču ili crtaču, film u kameri itd.) i sa kog se vrši unošenje pozicija u obliku dve koordinate **(x y)**. Za programera je sasvim nebitno da li je **lokator** miš, svetlosno pero, olovka grafičke table, ekran osetljiv na dodir, kursorski tasteri ili pak nešto sasvim deseto. Dokle god uređaj daje parove koordinata, ništa više o njemu nije potrebno znati.

Sledeći uređaj se naziva **procenjivač** i služi za unošenje skalarnih podataka koji se obično dobijaju procenom vrednosti nekog fizičkog parametra, bilo da se radi o mernom instrumentu sa A/D konvertorom, običnom potenciometru ili samo dva kursorska tastera koji podešavaju neki zamišljeni brojač. U sadašnjoj verziji GEM-a procenjivač daje vrednost u opsegu **[1..100]**, no nema nikakvih prepreka da se u budućnosti dobiju mnogo veći opsezi brojeva. Neki eksperimentalni grafički sistemi daju čak vrednosti u pokretnom zarezu sa 7 ili 15 tačnih cifara, a sve je to uglavnom stvar drajvera i konvencije koje se po potrebi uvek može promeniti.

Treći logički uređaj unosa je **birač** koji daje redni broj izabranog diskretnog stanja. To može biti redni broj tastera, prekidača u nekoj mreži, kružnog preklopnika itd. Bitno je da se ovde ne vrši nikakvo merenje niti pretvaranje analognog u digitalni signal, već se jednostavno vrši izbor jedne opcije. Ovo se može zgodno iskoristiti za rad sa malim spoljnim tastaturama. Kvalitetne grafičke table obično imaju takve priručne tastaturice za brzo zadavanje najbitnijih komandi.

Četvrti i poslednji logički uređaj unosa je **znakovni** uređaj koji daje nisku karaktera koji su obično u **ASCII** kodu, ali to nije obavezno. Standardno se za ovo unošenje

koristi tastaturu, no nema nikakvih prepreka da se u ovoj ulozi pojavi skaner sa ugrađenim računarom za prepoznavanje teksta ili čak uređaj za prepoznavanje govora. Priključivanje takvog uređaja uopšte neće smetati programima koji su do sada koristili tastaturu ukoliko je obezbeđen odgovarajući drajver.

Rad sa modelom radnih stanica u mnogome olakšava programerski posao. Ili bi jako barem trebalo da bude. Čitav sistem za sada uzima mnogo više memorije nego što je to razumno, program koji se brine o instalaciji i upravljanju drajverima, famozni GDOS, potpuno je nedokumentovan, a isto se odnosi i na drajvere. Zbog toga je vrlo teško izvršiti prilagođavanja za stalno dolazeća nova periferne uređaje, a to je bitan uslov efikasnosti čitavog sistema. Fontovi su poseban problem. Za novije i naprednije štampače je prava sreća pronaći po neki font iz prostog razloga što vrlo malo ljudi ima podatke o tome kako tačno treba da izgleda drajver koji će je fontove koristiti i kako fontova treba uređik da budu potpuno korektni i da se mogu slobodno povezivati. Zbog svih ih problema čitav sistem koriste samo oni koji rade profesionalne programe za vaniko tržište, a svi ostali koriste radne stanice samo onoliko koliko baš moraju da bi program korektno radio. Nekompletnost dokumentacije i česta promena verzija GDOS programa glavne su prepreke intenzivnijem korišćenju čitavog sistema radnih stanica u GEM-u. Za nove verzije atari mašina je obećano da će se ova stvar srediti.

Ekran

Ekran je svakako najvažniji izlazni i ulazni uređaj. VDI nama pod ekranom podrazumeva ono što se u TOS-u naziva konzolom, što će reći displej (kao izlazna jedinica) i tastatura s mišem (kao ulazna jedinica).

Kad su grafička operacija u pitanju, može se reći da je ekran jedan od najsposobnijih izlaznih uređaja na kome se u potpunosti izvršavaju sve grafičke operacije bez ikakvog ograničenja. Jedino se može zamariti da ugrađeni drajver ekrana pati od hronične neefikasnosti, što se kod jednostavnijih crteža ne primećuje, ali kod crteža sa par stotina osnovnih elemenata postaje jasno uočljivo. Za ekran se mogu otvarati **samo virtuelne radne stanice**. Fizička radna stanica je otvorena još u toku podizanja sistema i svaki pokušaj njenog ponovnog otvaranja odvešće vas ravno u pad sistema iz prostog razloga što će GEM u svim svojim sistemskim operacijama koristiti podatke o stanici koju je inicijalizacioni program otvorio, a takva stanica je uništena ponovnim „otvaranjem".

Za ekran kao uređaj, tj. za njegovu fizičku virtuelnu radnu stanicu **lokator je miš**, **procenjivač** čine **dva kursorska tastera**, **birač** čine **funkcijski tasteri** a **znakovni uređaj je tastatura**. Osim toga funkciju **lokatora** mogu vršiti i **tasteri kursorske grupe**.

Pri otvaranju virtuelne ekranske radne stanice možete kao i za sve druge uređaje odabrati NDC sistem, ali se programeri najčešće odlučuju za RC sistem koji obezbeđuje mnogo brži rad. NDC sistem se odabira samo u slučaju kad je prenosivost bitan zahtev programa i kad se žele izdavati slike na štampaču putem virtuelne radne stanice, tj. onda kad rasterska kopija ekrana (takozvani screen dump) ne zadovoljava željenu rezoluciju. Prenosivost programa se u svim slučajevima koji ne podrazumevaju neku vrstu CAD (Computer Aided Design) mogućnosti može rešiti i obično se rešava uvođenjem par konstanti koje određuju rezoluciju i čijom se izmenom lako vrši prilagođavanje za druge mašine.

Sve one koji vole da uprkos svim zabranama pročeprkaju direktno po ekranskoj memoriji jedino zanima kako je ta memorija organizovana i kako joj se može najbezbednije prići. Za pristup ekranskoj memoriji, koja zauzima 32 K, potrebna vam je adresa početka koju možete dobiti XBIOS funkcijom logbase( ) (Code 02), mada je preporučljivo *stvoriti sopstveni bafer* od 32 K u koma ćete konstruisati sliku i onda je jednostavno „prespojiti" na fizički izlaz.

Osnovna organizacija ekranske memorije je **red po red**, što znači da je u prvih 80 bajta (40 reči) bit mapirano zapisan prvi red od 640 tačaka, zatim je u sledećih 40 reči smešten drugi red itd. Čitava ekranska memorija **za visoku** (monohromnu) **rezoluciju** može se dakle definisati kao

WORD high_screen[400][40]

odakle se jasno vidi osnovna karakteristika koja određuje pristup tačkama. Osnovna jedinica kojoj možete pristupiti je **red od 16 tačaka**, koje su bit mapirano zapisane u jednoj reči. Ova grupa tačaka obično dobija poseban naziv **osnovna grupa**. U svim rezolucijama koje koriste boju i dalje se zadržava osnovna grupa od 16 tačaka, ali se ona predstavlja sa više reči.

U **srednjoj rezoluciji** jedan red i dalje ima 640 tačaka, ali sada ima samo 200 redova. U ovoj rezoluciji svakoj tački odgovaraju dva bita što daje mogućnost rada sa 4 boje. **Osnovna grupa se predstavlja sa dve reči**. Prva reč daje prvi, a druga daje drugi bit za svaku od 16 tačaka. U nešto starijoj terminologiji bi se reklo da svaka od ove dve reči određuje po jednu bit-ravan. U današnjoj situaciji kad se boja određuju zadavanjem fizičke vrednosti, tj. zadavanjem procentualnog udela R G i B komponenti (funkcijom **vs_color**), a boje se predstavljaju indeksima, nema nikakvog smisla govoriti o nekim bit-ravnima kad broj 0 1,2 ili 3 koji se dobija iz dva bita nije ništa drugo do puki indeks i nema nikakve direktne veze s bojom. Definicija ekranske memorije za **srednju rezoluciju** je

WORD med_screen [200][40][2]

U **niskoj rezoluciji** 4 bita određuju jednu tačku, pa postoji samo 320 tačaka u redu i 200 redova. **Osnovna grupa se predstavlja sa 4 reči**, što daje mogućnost rada sa 16 boja. Definicija ekranske memorije je

WORD low_screen [200][20][4]

Ukoliko niste specijalno željni podsećanja na stara 8-bitne dane direktnog rovanja po ekranskoj memoriji, koristite barem rutine A linije koje su uvek dovoljno brze a mnogo su lakše za korišćenje od direktnog preračunavanja tačaka i osnovnih grupa.

ST i PC računara. Ovakvi uređaji se koriste na moćnim grafičkim radnim stanicama da bi se slika od par hiljada tačaka po svakoj dimenziji direktno snimila na film i tako izbegli izuzetno skupi štampači. Na takvim mašinama se međutim ne koristi GEM već isključivo X Windows sa specijalizovanim programskim dodacima za rad sa snimačem koji sami prevazilaze moć personalnih računara.

U poslednjoj grupi su **funkcije prelaza za Metafile uređaj**. Ove su funkcije preko potrebne ako želite bilo kakav pristojniji crtež rađen u GEM-u spremiti na disk a da to ne bude u rasterskom obliku već u obliku komandi za crtanje osnovnih geometrijskih likova. Za sada su međutim date samo tri funkcije pa je sasvim jasno da ih ima dosta nedokumentovanih jer svi programi za objektivno crtanje kao i neki DTP (DeskTop Publishing) programi izvlače iz ovih fajlova ono što se sa samo tri date funkcije ni u snu ne bi moglo uraditi.

Sve u svemu, funkcije prelaza treba upoznati tek toliko da se ne zbunite kad naiđete na njih. Šta se korišćenja tiče, mala je gotovo nikakva korist od njihovog pozivanja.

Sve funkcije imaju zajednički glavni kod 05 a razlikuju se po pomoćnom kodu.

Kursor-adresivi ekran

(Cursor Addressable Screen Escapes)

Termin **alfa** ovde označava alfanumeričko prikazivanje tj. prikazivanje karaktera zadatih odgovarajućim (obično ASCII) kodovima. Razlikuje se od grafičkog prikazivanja po tome što je najmanja adresiva jedinica ćelija karaktera a ne tačka. Ovakvo adresiranje se naziva i **kursor adresiranje** ili adresiranje kursorom što nije sasvim precizno jer i u grafičkom načinu se vrši adresiranje kursorom. Jedina razlika je u veličini kursora tj. da li je on tačka (u grafičkom načinu) ili ćelija u alfa načinu.

VELIČINA ALFA-EKRANA
Code: 5—01 Inquire Alpha Character Cells

```
VOID vq_chcells (handle, rows, columns)
        WORD handle
        WORD *rows, *columns
```

U parametrima ***rows** i ***columns** funkcija vraća broj redova i kolona na alfa-ekranu. Ukoliko je vraćen broj -1 adresiranje ćelija nije moguće.

IZLAZAK IZ ALFA NAČINA
Code: 5—02 Exit Alpha Mode

```
VOID v_exit_cur (handle)
        WORD handle
```

Forsira prelazak ekrana iz alfa u grafički način prikazivanja.

ULAZAK U ALFA NAČIN
Code: 5—03 Enter Alpha Mode

```
VOID v_enter_cur (handle)
        WORD handle
```

Forsira prelazak ekrana iz grafičkog u alfa način prikazivanja.

POMERANJE ALFA KURSORA
Code: 5—04, 05, 06, 07

```
VOID v_curup (handle)
        WORD handle
VOID v_curdown (handle)
        WORD handle
VOID v_curright (handle)
        WORD handle
VOID v_curleft (handle)
        WORD handle
```

Ove četiri funkcije forsiraju pomeranje alfa kursora za jedno polje u jednom od četiri smera: **gore, dole, desno, levo**.

KURSOR NA POČETAK
Code: 5—08 Home Alpha Cursor

```
VOID v_curhome (handle)
        WORD handle
```

Forsira postavljanje alfa kursora u gornji levi ugao ekrana.

BRISANJE DO KRAJA EKRANA
Code: 5—09 Erase to End of Alpha Screen

```
VOID v_eeos (handle)
        WORD handle
```

Briše sav tekst od tekuće pozicije alfa kursora pa do kraja ekrana.

BRISANJE DO KRAJA REDA
Code: 5—0A Erase to End of Alpha Line

```
VOID v_eeol (handle)
        WORD handle
```

Briše sav tekst od tekuće pozicije alfa kursora pa do kraja reda.

POSTAVLJANJE ADRESE ALFA KURSORA
Code: 5—08 Set Alpha Cursor Address

```
VOID vs_curaddress (handle, row, column)
        WORD handle
        WORD row, column
```

Postavlja adresu alfa kursora. Korisnik to vidi kao postavljanje kursorske ćelije na ekranu na poziciju zadatu sa **row, column**.

ako se unos vrši **prekom**. Kodovi logičkih uređaja unose dati su u sledećoj tabeli

kod	**uređaj**
1	lokator
2	procenjivač
3	birač
4	znakovni uređaj

ISPITIVANJE TEKUĆEG FONTA
Code: 83 Inquire Font Info

```
VOID vqt_fontinfo (handle, minADE, maxADE,
        distances, maxwidth, effects)
    WORD handle;
    WORD minADE, *maxADE;
    WORD distances [5], *maxwidth;
    WORD effects [3];
```

Ovom se funkcijom dobijaju vrlo iscrpni podaci o tekućem fontu uključujući i sva specijalne efekte koji su tekući u vreme pozivanja funkcije. Parametri ***minADE** i ***maxADE** (ADE = ASCII Decimal Equivalent) daju opseg karaktera u fontu (ne mora svaki font imati ni 128 ni 256 karaktera). Niz **distances [5]** daje rastojanja glavnih linija karaktera od osnovne linije. Svaki element se odnosi na jednu liniju po sledećem rasporedu

element	**linija**
0	donja linija
1	linija spuštanja
2	linija polovine
3	linija podizanja
4	vršna linija

Na slici 40 vidite raspored glavnih linija

Slika 40

Promena širine i nagiba ćelije zbog specijalnih efekata dobijaju se u nizu **effects [3]**. Značenje elemenata ovog niza dato je sledećom tabelom

element	**značenje**
0	povećanje širine ćelije
1	levi ofset
2	desni ofset

Levi i desni ofset su različiti od nule samo za iskošena (ITALIC konstanta za **vst_effects**) slova. Na slici 30 vidite tačno značenja ovih brojeva

Slika 30

Funkcije prelaza
(Escape Functions)

Funkcije prelaza generalno služe za izvođenja brzog i jednostavnog prelaza između različitih stanja izlaznih uređaja. Svi korisnici računara koji su bili u (ne)prilici da podešavaju štampač slanjem gomila kontrolnih kodova, znaju da se praktično sve naredbe štampaču šalju tek iz ESC -karaktera koji jednostavno tera štampač da iz stanja štampanja pređe u stanje interpretiranja komandi. Naziv ove kategorije funkcija je međutim više istorijskog karaktera jer se u njoj stvarno nalaze sve one funkcije koje se nisu mogle „upakovati" u prethodne kategorije. Zbog toga se nemojte začuditi kad ovde naiđete i na funkcije izdavanja i na funkcije upita pa čak i na atributske funkcije

Ova se kategorija zvanično deli na sedam grupa no pojedine imaju samo par funkcija pa se toliko usitnjavanje čini nepotrebnim. Tek pojavom GEM-a 2.0. koji čak imati na raspolaganju tek ako uzmete neku MEGA ST mašinu počele su se i „manjinske" grupe popunjavati funkcijama. Iz naziva grupa bi se mogao steći utisak da funkcije prelaza služe za direktno upravljanje uređajima bez ikakvog posredovanja radnih stanica i sličnih softverskih modela. Sva ove funkcije međutim i dalje traže parametar **handle** tj. identifikacioni broj otvorene radne stanice. Ako raspolažete dovoljnim viškom slobodnog vremena možete pokušati da dajete „pogrešne" identifikacione brojeve tj. da pozivate ekranske funkcije za radnu stanicu štampača ili obratno. Čitav koncept funkcija prelaza deluje pomalo sumnjive u jednom tako razvijenom i tako „visokom" grafičkom programskom paketu kao što je GEM. Što ih više gledate to je jači utisak da se jednostavno radi o „dopunji" funkcije koje su tu ili rada radi ili iz njih tek treba da se razvije kompletna kategorija u nekoj budućoj verziji. Dok se to ne desi valja se pridržavati službene „raspodele" kako se ne bi napravila još veća zbrka.

U prvoj grupi su **funkcije za upravljanje kursor-adresivim ekranom** tj. ekranom koji može prikazivati samo alfanumeričke podatke kao što to i dan danas čini većina terminala po računskim centrima. Sva su ta funkcije u GEM jednostavno zalutale. Ko želi raditi u alfanumeričkom načinu tj. odreći se bilo kakve grafike uopšte ne uključuje GEM već jednostavno radi pod „golim" operativnim sistemom (TOS, MSDOS itd.) gde se sva kontrola izvodi mnogo jednostavnije

U drugoj grupi su **opšte ekranske funkcije prelaza** koje već obavljaju i neki konstan posao kao što je na primer izdavanje „čvrste" kopije ekrana na štampač (među programerima poznatije kao **hardcopy** ili **screen-dump**)

Treću grupu čine **funkcija prelaza štampača** koje povremeno mogu biti vrlo upotrebljive a jednog dana bi se mogla razviti u samostalnu kategoriju. Da ne bude zabune ove funkcije ne šalju štampaču ESC-sekvence već obavljaju neke kućne poslove vezane za štampanje rasterski zadatih slika, pomeranje papira itd.

Četvrta grupa nosi zvučni naziv **uređajno-zavisnih prelaza** a ima svega jednu funkciju. Isti je slučaj i sa sledećom grupom koja treba da sadrži **funkcije prelaza grafičke table** a ne sadrži nijednu jedinu funkciju. Ove dve grupe su u GEM u 2.0 popunjena sa po nekoliko funkcija što je još uvek male i sumnjiva podrška ovim uređajima pa se programeri radije odlučuju da kad već otvaraju radnu stanicu za ove uređaje koriste samo funkcija prethodnih grupa za rad sa njima

U šestoj grupi su **funkcije prelaza za animač filma** za koje samo autori znaju čemu služe na mašinama popul

Fontovi

Fontovi su, kao i ekranska memorija, organizovani u **red po red** poretku, s tom razlikom da ovde nema nikakvih osnovnih grupa. Zamislite da ste sve karaktere fonta napisali u jednom redu, pa potom nekim skanerom „skinuli" prvu liniju, potom drugu, treću itd. Upravo je tako organizovan font. Ako su svi karakteri široki, na primer 8 bita, onda drugi karakter počinje od devetog bita, treći od osamnaestog itd. Jedini je uslov da svaki novi red počinje na granici cele reči. Time se značajno štedi memorija ali i usporava ispis karaktera. Zbog toga su sistemski karakteri organizovani sa matricama od 8×8 i 8×16 tačaka, pa nema nikakvog preskakanja granica bajta odnosno reči. Postoji i takozvani sitni font sa matricom 6×8, no on se koristi samo za tekst u ikonama

Font se sastoji od četiri dela. To su **zaglavlje (header)**, **tabela horizontalnih ofseta (Horizontal Offset Table — HOT)**, **tabela znakovnih ofseta (Character Offset Table — COT)** i **sami podaci o znacima (character data)**, tj. blok linija smeštenih kao bit-mapirane reči

Zaglavlje fonta je dugačko 88 bajtova i sadrži sve podatke koji su sistemu potrebni da bi mogao da učita i koristi font. Izuzetak je sistemski font koji se može učitati „go" tj. blok od 2048 ili 4096 bajtova u kome je 8 linija sa po 256×8 ili 256×16 tačaka. Zaglavlje fonta se može definisati kao

```
TYPEDEF STRUCT
WORD ID             /* identifikacioni broj */
WORD size           /* visina u tačkama (1 tačka=1/72 inča) */
CHAR name [32]      /* ime fonta */
/* font ne mora imati svih 256 znakova ali svi znaci koji postoje u fontu moraju imati neprekinut niz kodova */
WORD min_char       /* najniži ASCII kod znaka u fontu */
WORD max_char       /* najviši ASCII kod znaka u fontu */
/* rastojanja od osnovne linije */
WORD top_line       /* vršna linija */
WORD asc_line       /* linija podizanja */
WORD half_line      /* linija polovine */
WORD desc_line      /* linija spuštanja */
WORD bot_line       /* donja linija */
/* ***** specijalni efekti ***** */
WORD width_char     /* širina najvećeg znaka */
WORD width_cell     /* širina najveće ćelije */
WORD l_offset       /* levi i */
WORD r_offset       /*      desni ofset (vidi vqt_fontinfo) */
WORD bold           /* zadebljanje znaka u tačkama */
WORD under          /* debljina linija za podvlačenje */
WORD light          /* maska za izbleđivanje fonta */
WORD italic         /* maska za stvaranja iskošenog (ital[illegible] */
WORD FLAGS          /* bit mapirano
                    bit-0=1 sistemski font
                    0 nije sistemski font
                    bit 1=1 postoji HOT
                    0 ne postoji HOT
                    bit 2=1 brojevi u Motorola formatu ($ABCD=AB CD)
                    0 brojevi u Intel formatu ($ABCD=CD AB)
                    bit 3=1 neproporcionalan font
                    0 proporcionalan font */
LONG HOT_addr       /* adresa tabele horizontalnih ofseta */
LONG COT_addr       /* adresa tabele znakovnih ofseta */
LONG DATA_addr      /* adresa podataka tj. bloka bit linija */
WORD form_w         /* dužina jedne linije u bajtima */
WORD form_h         /* broj linija (za štampač je obično
                    dvostruko veći od visine u tačkama) */
LONG next_font      /* adresa sledećeg fonta — postavlja drajver */
FONT HEADER
```

HOT ne mora biti prisutan a kad jeste definicija mu je

WORD HOT [max_char — min_char+1]

I svaki element sadrži ofset koji se podaje poziciji znaka pra nego što se počne s iscrtavanjem. Ako je ofset negativan može se koristiti za štampanje akcenata.

COT mora uvek biti prisutan definicija mu je

WORD COT [max_char — min_char+1]

I svaki element daje poziciju početka bitova koji definišu određeni znak. Pozicija je n odnosu na početak linije i koristi se da bi se uopšte moglo pronaći mesto znaka. Isti ofset važi za sve linije jačaka. Ako je ASCII kod znaka **char** onda je njegov ofset **COT (char — min_char).**

Bit-2 elementa **FLAG** odnosi se **na sve brojeve** kad n **zaglavlju** tako i n **HOT-u** i **COT-u**. Drajver prvo ispituje ovaj bit pa onda počinje da uzima podatke iz zaglavlja.

Line Below 24 – **sredne funkcije**

Funkcije raznih kategorija su često toliko povezane da ih je gotovo besmisleno koristiti zasebno. Pre zdavanja nekog grafičkog lika na ekran treba mu podesiti atribute a pre toga treba utvrditi tekuću vrednost atributa itd.

Zbog toga će n nastavku biti date tabele srodnih funkcija pomoću kojih ćete se lakše snalaziti u pravom moru atributskih upitnih itd. operacija.

Funkcije upita i atributske funkcije

vql_atributes	**vqm_atributes**
vsl_type	vsm_type
vsl_color	vsm_color
vswr_mode	vswr_mode
vsl_width	vsm_height
vsl_ends	vsf_perimeter

vqf_attributes	**vqt_attributes**
vsf_interior	vst_font
vsf_color	vst_color
vsf_style	vst_rotation
vswr_mode	vst_alignment
	vswr_mode
	vst_height

Funkcije izdavanja i atributske funkcije

v_pline	vst_height
vswr_mode	vst_point
vsl_type	vst_rotation
vsl_udsty	vst_font
vsl_width	vst_color
vsl_color	vst_effects
vsl_ends	vst_alignment

v_pmarker	**v_fillarea**
vswr_mode	vswr_mode
vsm_type	vsf_interior
vsm_height	vsf_style
vsm_color	vsf_color
	vsf_perimeter
	vsf_updat

v_gtext	**v_contourfill**
vswr_mode	**vr_recfl**
vst_height	vswr_mode
vst_point	vsf_interior
vst_rotation	vsf_style
vst_font	vsf_color
vst_color	vsf_updaj
vst_effects	
vst_alignment	

GDP funkcije

Upotreba atributa za GDP funkcije je prilično inificirana pa će, radi uštede prostora, biti uvedene dva grupe atributa **LINE** **FILL** koje se najčešće koriste tako da će kasnije biti navedeno samo njihovo ime kod funkcija koje ih koriste.

LINE	**FILL**
vswr_mode	vswr_mode
vsl_type	vsf_interior
vsl_udsty	vsf_style
vsl_width	vsf_color
vsl_color	vsf_perimeter
vsl_ends	vsf_updat

Pri korišćenju ovih tabela treba imati ne umu da postoji posredan uticaj nekih funkcija iz LINE na funkcije koje su pod nlcajem grupe FILL. Uvek kad je funkcijom **vsf_perimeter** postavljeno stanje n kom se vrši iscrtavanje ruba koji nije ništa drugo do linija, atributi linije utiču ne proces iscrtavanja.

v_bar	**v_ellpie**
FILL	FILL
	v_ellipse
	FILL
v_arc	**v_rbox**
LINE	LINE
v_pie	
FILL	**v_rfbox**
v_circle	FILL
FILL	
	v_justified
v_ellarc	vswr_mode
LINE	

Funkcije VDI-a

Funkcije VDI-a se mogu grupisati u sedam kategorija. Za korišćenje VDI funkcija potrebno je međutim prvo „ba-

OPSEG TEKSTA
Code: 74 Inquire Text Extent

```
VOID vqt_extent (handle string extent)
        WORD handle
        CHAR string
        WORD extent [8]
```

Vraća podatke n pravougaoniku koji bi zauzimao tekst kad bi bio izdat funkcijom **v_gtext**. Pošto tekst može biti rotiran potrebne su koordinate sva četri temena pravougaonika. One se vraćaju kao elementi niza **extent [8]** po sledećem rasporedu

(x, y)	teme
0 1	donje levo
2 3	donje desno
4 5	gornje desno
6 7	gornje levo

ŠIRINA ĆELIJE KARAKTERA
Code: 75 Inquire Character Cell Width

```
WORD vqt_width (handle character cell_width
                left_delta, right_delta)
        WORD handle
        CHAR character
        WORD *cell_width
        WORD left_delta
        WORD *right_delta
```

Ovom se funkcijom dobijaju detaljni podaci o ćeliji jednog karaktera zadatog parametrom **character** tekućeg fonta. Funkcija ne uračunava nikakve specijalne efekte a sve podatke daje n tekućem koordinatnom sistemu (NDC ili RC). Na slici 20 vidite odnos širine samog karaktera i širine njegove ćelije.

Slika 20

Parametri ***left_delta** i ***right_delta** daju rastojanje od leve odnosno desne ivice ćelije do samog karaktera.

IME I INDEKS FONTA
Code: 82 Inquire Font Name and Index

```
WORD vqt_name (handle element_num name)
        WORD handle
        WORD element_num
        CHAR name []
```

U ASSIGN SYS fajlu se nalaze spiskovi raspoloživih fontova za svaki uređaj. Ovom funkcijom se iz rednog broja fonta u ASSIGN SYS fajlu zadatog parametrom **element_num**, dobija njegovo ime i indeks koji je potreban kao parametar za funkciju **vst_font**. Ime se dobije n stringu **name** a indeks se vraća kao vrednost funkcije.

MOZAIK OD ĆELIJA
Code: 18 Inquire Cell Array

```
VOID vq_cellarray (handle pxyarray
                   row_length num_rows,
                   el_used rows_used
                   status colarray)
        WORD handle
        WORD pxyarray []
        WORD row_length num_rows
        WORD *el_used *rows_used
        WORD *status
        WORD *colarray
```

Ova je funkcija „par" sa funkcijom **v_cellarray** za koju je već rečeno da je prilično nepotrebna. Za ovu se funkciju čak može reći i da je problematične.

Na zadatoj radnoj stanici se parametrom **pxyarray [4]**, određuje pravougaonik koordinatama gornjeg levog i donjeg desnog temena po standardnom formatu

(x1 y1) – (pxyarray **[0]**, pxyarray **[1]**)
(x2, y2) – (pxyarray **[2]**, pxyarray **[3]**)

Ovaj se pravougaonik deli na **num_rows** redova i **el_used** kolona a od VDI a se traži da proizvoljni deo radne površine (obično je to ekran) smatra za mozaik od ćelija te da u nizu **colarray** [] (koji može biti jedno il dvo-dimenzionalan) vrati podatke o bojama svake ćelije a u parametrima ***el_used** ***rows_used** stvarni broj elemenata u redu i koloni (koji je jednak onom koji je zadat ili broju tačaka ovisno o tome koji je od ta dva broja manji). Potom šalje ovo primeniti na deo ekrana n kome ste iscrtali razne geometrijske likove n raznim bojama!

Nigde nije dat nikakav podatak o tome po kom se kriterijumu određuje boja ćelije n kojoj se nalaze tačke sa različitim bojama. Jedina prihvatljiva primena ova funkcije je očitavanje ćelija mozaika ispisanog v **colarray** funkcijom je očitavanja boja svake tačke zadatog pravougaonika. Postavlja se međutim pitanje čemu sve to? Analiziranje oblasti ekrana tačku po tačku je posao sa kojim i UNIX radne stanice teško izlaze na kraj a očitavanje boja mozaičkih ćelija nema baš nikakvu realnu svrhu sem isprobavanja. Crtanje mozaika se i ovako praktično nikada ne koristi a i tamo gde se koristi služi samo za prikazivanje pedetalja koji su već nmenčlu obrađeni i spremljeni n neke nizove ili čak fajlove pa ih je uvek lakše i brže odatle uzeti.

Sve u svemu pred vama je funkcija za one se viškom živaca i vremena za bacanje.

NAČIN UNOŠENJA
Code: 73 Inquire Input Mode

```
VOID vqin_mode (handle dew type input_mode)
        WORD handle
        WORD dew_type
        WORD *input_mode
```

Za zadatu radnu stanicu i zadati logički uređaj unosa, čiji se kod zadaje parametrom **dev_type**, vraća n parametru ***input_mode** broj **1** ako se unos vrši **zahtevom** a **2**

Dodatni podaci

Termin **alfa** ovde označava alfanumeričko prikazivanje tj. prikazivanje karaktera zadatih odgovarajućim (obično ASCII) kodovima. Razlikuje se od grafičkog prikazivanja po tome što je najmanja adresiva jedinica ćelija karaktera a ne tačka.

work_out [0]	tip ekrana: 0 nema ekranskog uređaja 1 odvojeni alfa i grafički kontroler odvojeni ekrani 2 odvojeni alfa i grafički kontroler zajednički ekran ekran 3 zajednički alfa i grafički kontroler odvojene memorije slike 4 zajednički alfa grafički kontroler zajednička memorija slike
work_out [1]	broj različitih boja pozadine
work_out [2]	podržani specijalni efekti (kodirano kao za **vst-effects**)
work_out [3]	podržano skaliranje rastera (0-ne 1-da)
work_out [4]	broj bit-ravni
work_out [5]	tabele za RGB-specifikaciju boja (0-ne 1-da)
work_out [6]	brzina (broj 16×16 rasterskih operacija u sec)
work_out [7]	mogućnost popunjavanja likova bojom (0-ne 1-da)
work_out [8]	mogućnost rotacije teksta 0- ne 1 samo za 90 stepeni 2- svi uglovi
work_out [9]	broj raspoloživih načina ispisivanje
work_out [10]	raspoloživi načini unošenja 0- nijedan 1 zahtevom 2 zahtevom i probom
work_out [11]	mogućnost podešavanja teksta (0-ne, 1-da)
work_out [12]	promena boje promenom pera (0-ne 1-da)
work_out [13]	mogu izdavanja „rastezućih" linija 0- nije moguće 1 samo prave linije 2 prave linije i pravougaonici
work_out [14]	najveći broj tačaka u izlomljenoj liniji grupi markera i popunjenom poligonu (1 — nema granice)
work_out [15]	najveća dužina **intin** niza (1 – nema granice)
work_out [16]	broj dugmadi da mišu
work_out [17]	važenje tipa za debele linije (0-ne 1-da)
work_out [18]	važenje načina ispisivanja za debele linije (0-ne 1-da)
work_out [19]	work_out 56 rezervisano – 0

Vraća tekuće vrednosti atributa popunjavanja u nizu **attrib 5** po sledećem rasporedu

attrib [0]	način popunjavanja
attrib [1]	boja za popunjavanje
attrib [2]	indeks uzorka
attrib [3]	način pisanja
attrib [4]	vidljivost ruba

ATRIBUTI TEKSTA

Code: 26 Inquire Text Attributes

```
VOID vqt_attributes (handle  attrib)
        WORD  handle
        WORD  attrib [10]
```

Vraća tekuće vrednosti atributa teksta u nizu **attrib 10** po sledećem rasporedu

attrib [0]	indeks fonta
attrib [1]	boja teksta
attrib [2]	nagib teksta
attrib [3]	horizontalno podešavanje
attrib [4]	vertikalno podešavanje
attrib [5]	način pisanja
attrib [6]	širina karaktera
attrib [7]	visina karaktera
attrib [8]	širina ćelije
attrib [9]	visina ćelije

ciri" nekoliko kilobajta za definisanje nizova preko kojih se prenose podaci. Od pojave GEM-a svi se programeri čude kako je neko uopšte mogao smisliti tako loš način povezivanja. Funkcije definisane u C-u ili na kom drugom programskom jeziku već imaju parametre koji se potom moraju prepisivati u odgovarajuće nizove, pa se tek potom poziva GEM. Time se nepotrebno gubi i vreme i prostor no šta je tu je. Neophodni nizovi su

```
WORD      contrl [12]
WORD      intin [128]
WORD      ptsin [128]
WORD      intout [128]
WORD      ptsout [128]
WORD   work_in [11]
WORD   work_out [57]
WORD   pxyarray [10]
```

Raspored funkcija po kategorijama nije baš uvek najlogičniji (često par funkcije „zaluta" u drugu kategoriju), no kako se radi o grafičkom sistemu namenjenom korisnicima više različitih tipova mašina „propisi" su vrlo strogi. Stoga je u potpunosti poštovana originalna specifikacija koju je dao Digital Research.

Prvu kategoriju čine funkcije za „kućne poslove" tj. za otvaranje, zatvaranje i brisanje realnih i virtuelnih radnih stanica, za učitavanje i brisanje fontova te za određivanje dozvoljene grafičke površine. Službeni naziv im je **kontrolne funkcije**.

U drugoj kategoriji su funkcije za crtanje osnovnih geometrijskih likova, popunjenih i nepopunjenih likova, otvorenih i zatvorenih izlomljenih linija i teksta. Tekst se naime crta kao i svaki drugi grafički simbol. To omogućava jednostavno menjanje oblika, veličine i nagiba teksta u prilično velikom rasponu. Službeni naziv funkcija ova kategorije je **funkcije izdavanja.**

Funkcije iz treće kategorije se službeno nazivaju **atributskim**, što rečito govori i o njihovom delovanju. Ovim funkcijama se naime menjaju atributi osnovnih geometrijskih likova (koje crtaju funkcije iz prethodne kategorije). I to ne samo boja i debljina linija već i uzorci za popunjavanje likova, oblik slova, nagib linije po kojoj se slova ispisuju itd. Pametna upotreba ovih funkcija je ključ dobrog grafičkog izgleda slike na ekranu, štampaču, ploteru itd.

Četvrtu kategoriju čine samo četiri funkcije koje se službeno nazivaju **rasterskim** i služe za pomeranje velikih blokova „tačaka" s jednog dela ekrana na drugi. Ove funkcije imaju smisla praktično samo za ekran i obično se koriste za skrolovanje teksta i slika u prozorima.

Petu kategoriju čine funkcije za unošenja podataka, pa im je stoga službeno ime **funkcije unosa.** Postoje četiri osnovna načina kojima korisnik može uneti neki podatak preko VDI sistema. Osim toga, nekim značajnim događajima vezanim za unošenje podataka mogu se dodeliti korisnička mašinske rutine čime se daje mogućnost dodavanja novih metoda i uređaja za unošenje podataka.

Šestu kategoriju čine funkcija kojima se utvrđuje trenutno stanje atributa (koji se postavljaju funkcijama prethodne grupe) i svih parametara otvorene radne stanice. Službeni naziv za njih je **funkcije upita.**

Sedmu kategoriju čine **funkcije prelaza,** poznatije među programerima po svom originalnom nazivu **escape.** Ove funkcije su kompletno zavisne od uređaja koji je predstavljen radnom stanicom. Svrha im je kako promena režima rada uređaja (grafički, tekstualni itd.) tako i izvršavanje određenih sitnih operacije na uređaju (pomeranje kursora, upit o stanju grafička table, štampača itd.). Treba među tim praviti razliku između ovih funkcija i escape-karaktera koje podržava TOS simulirajući VT 52 terminal.

Kontrolne funkcije

(CONTROL functions)

U ovoj kategoriji se nalazi devet funkcija kojima se otvaraju, zatvaraju, brišu i ažuriraju radne stanice, određuje prostor za izdavanje grafičkih simbola te postavljaju i uništavaju fontovi.

GEM VDI svaki ulazni ili izlazni uređaj smatra **radnom stanicom** koja se mora otvoriti da bi se s njom komuniciralo. To otvaranje podrazumeva učitavanje upravljačkog programa koji treba da upravlja uređajem. Ovi programi se stoga i nazivaju „vozačima" tj. drajverima (transkripcija engleske reči **driver**), a prepoznajete ih po ekstenziji **SYS.** O svima njima brine **GDOS,** deo GEM-a koji stavljate u AUTO folder (kao da se izvršava pri podizanju mašine) i ostaje rezidentan. Spisak svih drajvera, sa identifikacionim brojevima uređaja kojima upravljaju mora da se nalazi u **ASSIGN.SYS** datoteci na startnom (boot) disku. Identifikacioni brojevi uređaja su

1—10	Ekran (Screen)
11—20	Crtač (Plotter)
21—30	Štampač (Printer)
31—40	Metafile
41—50	Kamera (Camera)
51—60	Tabla (tablet)

Uređaj koji se naziva **Metafile** nije nikakav fizički uređaj. Njegov upravljački program otvara fajl na disku u koji se upisuju svi grafički objekti koje program hoće da „nacrta" na ovom uređaju. Na taj način se elegantno rešava spremanje i učitavanje crteža koji se sastoje od primitivnih geometrijskih likova. Tako rade svi „objektni" programi za crtanje. Rezultat delovanja **Metafile** uređaja vidi se na disku kao običan fajl sa ekstenzijom **GEM.** Najbolja stvar kod ovih fajlova je to što se mogu prenositi i na druge mašine koje rade pod GEM-om. Tako će PC bez problema primiti GEM fajl urađen na ST i obratno.

Po otvaranju fizičke radne stanice treba otvoriti i virtuelnu. Virtuelna radna stanica je kao i fizička softverski model. Za razliku od fizičke radne stanice, čija je svrha sirnvanje upravljačkih hardverski zavisnih delova GEM-a, svrha virtuelne radne stanice je dodeljivanje više radnih okolina istom uređaju. Pošto je GEM zamišljen kao višeprogramski grafički sistem, potreba za višestrukim korišćenjem istog uređaja je sasvim jasna. Međutim, čak i jedan program može izvući mnoge koristi od rada sa više virtuelnih stanica na istom uređaju, naročito ako je taj uređaj ekran. Svakoj fizičkoj radnoj stanici se po otvaranju dodeljuje referentni broj **handle**, koji se mora predati pri otvaranju virtuelne stanice koja takođe dobija svoj **handle.**

Ovaj broj je jedini način razlikovanja radnih stanica u svim kasnijim operacijama pa ga treba dobro čuvati u varijabli koju nikada ni jedna naredba neće „dirati".

OTVARANJE RADNE STANICE
Code: 01 Open Workstation

```
VOID v_opnwk (work_in handle work_out)
        WORK work_in[]
        WORD *handle
        WORD work_out[]
```

Otvara fizičku radnu stanicu čiji se atributi zadaju elementima niza **work_in**:

work_in [0]	identifikacioni broj uređaja
work_in [1]	tip linije
work_in [2]	boja izlomljene linije
work_in [3]	tip markera
work_in [4]	boja markera
work_in [5]	font
work_in [6]	boja slova
work_in [7]	način popunjavanja likova
work_in [8]	tip uzorka za popunjavanje
work_in [9]	boja uzorka
work_in [10]	**0 NDC 2 RC sistem**

Svaka radna stanica može koristiti jedan od dva koordinatna sistema. Prvi se naziva normalizovanim (**NDC — Normalized Device Coordinates**) a drugi rasterskim (**RC — Raster Coordinates**). Normalizovani sistem je zamišljen kao univerzalni i hardverski nezavisan koordinatni sistem sa početkom u donjem levom uglu i rasponom koordinata **[0..32767]**. Ovaj sistem se međutim retko koristi jer zahteva suviše preračunavanja pri generisanju slike a to bitno usporava rad upravljačkog programa, naročito ako je uređaj ekran. Rasterski sistem koristi stvarne koordinate uređaja (na primer 640×400 ili 320×200) pa je zato crtanje u njemu mnogo brže ali i teže prenosivo na druge uređaje.

Podaci o otvorenoj radnoj stanici vraćaju se u 57 elemenata niza **work out**. Navođenje značenja svih elemenata bi oduzelo mnogo prostora, a njih i ovako koriste samo autori vrhunskih profesionalnih programa pa ih se ovde može preskočiti. Značenje elemenata ovog niza biće objašnjeno kod funkcija **vq extnd** koja pripada kategoriji funkcija upita.

Najvažnija uputstvo za korišćenje ove funkcije je *ne mojte je koristiti ukoliko to nije baš neophodno*. Radna stanica za ekran je otvorena još u vreme podizanja sistema a njeno ponovno otvaranje je ne samo besmisleno već i pogubno jer često izaziva pad sistema. Otvaranje stanice za neki drugi uređaj verovatno vam nikada neće ni trebati. Za štampanje običnog teksta ili crteža sa ekrana koristiće te mnogo češće standardne ESC sekvence vašeg štampača. Samo u slučaju da radite program koji želite da prenesete na druge mašine ili želite da koristite standardni VDI-drajver za svoj štampač *morate otvoriti radnu stanicu za štampač*. Program koji to radi troše međutim dosta sistemske memorije. Jedno od standardnih rešenja je jednostavno spremanje rasterske slike u fajl za kasnije štampanje nekim od standardnih programa za crtanje. Uopšte govoreći, dokle god ste zadovoljni rasterskom slikom sa ekrana možete mirne duše slati sliku svom štampaču standardnim TOS-funkcijama bajt po bajt iz ekranske memorije.

ZATVARANJE RADNE STANICE
Code: 02 Close Workstation

```
VOID v_clswk(handle)
        WORD handle
```

Zatvara fizičku radnu stanicu. Ukoliko je fizička radna stanica crtač, pisač ili kamera, izvršava se ažuriranje i slanje svih eventualno zaostalih podataka. Ekran se vraća u znakovni mod a lati pridružen Metfile uređaju se zatvara i ostavlja na disku.

Pre zatvaranja fizičke radne stanice moraju se zatvoriti sve virtualne koje joj pripadaju. Zatvoriti se može samo ona radna stanica koja je pre toga otvorena.

OTVARANJE VIRTUALNE RADNE STANICE
Code: 64 Open Virtual Workstation

```
VOID v_opnvwk (work_in handle work_out)
        WORD work_in[]
        WORD *handle
        WORD work_out[]
```

Otvara virtualnu radnu stanicu. Parametri koji se predaju i dobijaju od ove funkcije imaju značenje identično onom kod otvaranja fizičke radne stanice uz jednu razliku. Varijabla **handle**, čija se adresa predaje kao parametar, mora sadržati identifikacioni broj fizičke radne stanice za koju se otvara virtualna. Ukoliko imate nameru da otvorite više virtualnih stanica, sačuvajte ovaj identifikacioni broj u nekoj drugoj varijabli.

Osnovno pravilo za ovu funkciju je: MORA SE POZVATI pre bilo kakvog pokušaja izdavanja bilo kakvih grafičkih elemenata. Pošto se sve izlazno-ulazne operacije izvode isključivo preko virtualnih radnih stanica, to su njeni atributi (boja, oblik linija, teksta itd.) važeći. Atributi fizičke radne stanice praktično i ne dolaze do izražaja osim u nekim posebnim situacijama.

ZATVARANJE VIRTUALNE RADNE STANICE
Code: 65 Close Virtual Workstation

```
VOID v_clsvwk (handle)
        WORD handle
```

Zatvara virtualnu radnu stanicu. Sve ono što je rečeno za zatvaranja fizičke radne stanice važi i za zatvaranja virtualne, s tom razlikom da se ne vrši nikakvo ažuriranje fizičkog uređaja. Zatvaranjem ekranske virtualne radne stanice nećete dakle ni izbrisati ekran niti ga vratiti u karakterski mod. Jednostavno rečeno neće se ništa dogoditi sem što **handle** više neće biti važeći.

Standardni podaci

work_out [0]	najveća adresiva tačka na x-osi (639 za visoku rezoluciju 319 za nisku)
work_out [1]	najveća adresiva tačka na y-osi (399 za visoku rezoluciju 199 za srednju nisku)
work_out [2]	0 uređaj daje precizno skaliranu sliku (štampač, crtač...) 1 ne daje (sn mac filma)
work_out [3]	širina tačke u mikronima
work_out [4]	visina tačke u mikronima
work_out [5]	broj različitih visina karaktera (0-sve visina)
work_out [6]	broj tipova linija
work_out [7]	broj različitih debljina linija (0 — sve debljine)
work_out [8]	broj tipova markera
work_out [9]	broj različitih visina markera (0 — sve visina)
work_out [10]	broj različitih fontova
work_out [11]	broj različitih uzoraka za popunjavanje
work_out [12]	broj različitih šrafura
work_out [13]	broj istovremeno prikazivih boja
work_out [14]	broj GDP a (osnovni geometrijski likovi)
work_out [15]	work_out 24 spisak podržanih GDP a. Svaki element sadrži pomoćni kod jedne od 10 funkcija opisanih u poglavlju o funkcijama izdavanja. Ako je podržano manje od 10 GDP-a spisak se završava brojem −1
work_out [25]	work_out 34 spisak atributa koji utiču na pojedinačne GDP. Svaki element odgovara jednom GDP u (25—15 26—16 ...). Elementi sadrže kodove grupa atributa 0 atributi linije 1 atributi markera 2 atributi teksta 3 atributi popunjavanja 4 nema atributa
work_out [35]	mogućnost rada sa bojom (0 ne 1-da)
work_out [36]	mogućnost rotacija teksta (0-ne 1-da)
work_out [37]	mogućnost popunjavanja likova (0 ne 1-da)
work_out [38]	mogućnost izdavanja mozaika (0-ne 1-da)
work_out [39]	broj raspoloživih boja (2 monohromno 0 više od 32767)
work_out [40]	broj lokatora (1 samo tastatura)
work_out [41]	broj procenjivača (1-samo tastatura)
work_out [42]	broj birača (1 samo tastatura)
work_out [43]	broj znakovnih uređaja (za unos stringa)
work_out [44]	tip radne stanice 0 samo izlazna 1 samo ulazna 2 ulazno/izlazna 3 rezervisano 4 Metafile uređaj
work_out [45]	najmanja širina karaktera po x-osi
work_out [46]	najmanja visina karaktera po y osi
work_out [47]	maksimalna širina karaktera po x-osi
work_out [48]	najveća visina karaktera po y osi
work_out [49]	najmanja debljina linija po x osi
work_out [50]	0 (ne koristi se)
work_out [51]	najveća debljina linija po x-osi
work_out [52]	0 (ne koristi se)
work_out [53]	najmanja širina markera po x osi
work_out [54]	najmanja visina markera po y osi
work_out [55]	najveća širina markera po x osi
work_out [56]	najveća visina markera po y osi

Parametar *pstatus sadrži bit mapiran podatak o pritisnutim dugmadima (najviše ih može biti 16). Tako za standardni miš postoje četiri mogućnosti

0 (00)	ništa nije pritisnuto
1 (01)	levo dugme
2 (02)	desno dugme
3 (03)	obe dugmeta

STANJE TASTATURE
Code: 80 Sample Keyboard State

VOID vq_key_s (handle, pstatus)
WORD handle
WORD *pstatus

Ispituje trenutno stanje specijalnih tastera na tastaturi. Specijalni tasteri su **Shift, Control, Alternate**. To su tasteri čiji pritisak ne izaziva tastaturni prekid već samo modifikuje kod drugih pritisnutih tastera. Postoje u stvari dva Shift tastera, levi i desni. Skraćene oznake za specijalne tastere su **Ctrl** za Control i **Alt** za Alternate.

Parametar ***pstatus** sadrži bit-mapirano stanje sva četiri tastera. Osnovna stanja su

0 (0000)	ništa nije pritisnuto
1 (0001)	pritisnut R-Shift
2 (0010)	pritisnut L-Shift
4 (0100)	pritisnut Ctrl
8 (1000)	pritisnut Alt

i iz njih može nastati 16 kombinacija ali se obično pritisak na bilo koji Shift taster smatra jednim stanjem pa se broj kombinacija svodi na 8.

Funkcije upita
(Inquire functions)

Ova kategorija sadrži 12 funkcija. Sve su one namenjene uzimanju podataka o trenutnom stanju radne stanice i resursima kojima ona raspolaže. Praktično sve atributske funkcije imaju svoje parove u funkcijama upita a tu su i funkcije koje snabdevaju program čitavom gomilom podataka o najsitnijim detaljima vezanim za otvorenu radnu stanicu.

PROŠIRENI UPIT
Code: 66 Extended Inquire

VOID vq_extnd (handle, owflag, work_out)
WORD handle
WORD owflag
WORD work_out []

Vraća kompletnu informaciju o otvorenoj radnoj stanici. Ukoliko je parametar **owflag**=0, niz **work_out 57** sadrži sve one standardne podatke koje funkcije **v_opnwk** i **v_opnvwk** vraćaju. U suprotnom će ovaj niz u svojih prvih 19 elemenata sadržati dodatne podatke o grafičkim mogućnostima radne stanice.

ISPITIVANJE PALETE
Code: 0A Inquire Color Representation

WORD vq_color (handle, color_index, set_flag, rgb)
WORD handle
WORD color_index
WORD set_flag
WORD rgb [3]

Vraća fizičku vrednost boje za indeks zadat parametrom **color_indeks**. Ukoliko je **set_flag**=0, vraćaju se zahtevani intenziteti hromatskih komponenti a u suprotnom se vraćaju realni intenziteti tj. najbliži raspoloživi intenziteti koji su često skoro identični sa zahtevanim.

Elementi niza **rgb [3]** sadrže intenzitete hromatskih komponenti po sledećem rasporedu

rgb_in [0]	R
rgb_in [1]	G
rgb_in [2]	B

ATRIBUTI LINIJE
Code: 23 Inquire Line Attributes

VOID vql_attributes (handle, attrib)
WORD handle
WORD attrib [6]

Vraća tekuće vrednosti atributa linije u nizu **attrib 6** po sledećem rasporedu

attrib [0]	tip linije
attrib [1]	boja linije
attrib [2]	način pisanja
attrib [3]	debljina linije
attrib [4]	početak linije
attrib [5]	završetak linije

ATRIBUTI MARKERA
Code: 24 Inquire Marker Attributes

VOID vqm_attributes (handle, attrib)
WORD handle
WORD attrib [5]

Vraća tekuće vrednosti atributa markera u nizu **attrib 5** po sledećem rasporedu

attrib [0]	tip markera
attrib [1]	boja markera
attrib [2]	način pisanja
attrib [3]	visina markera
attrib [4]	širina markera

ATRIBUTI POPUNJAVANJA
Code: 25 Inquire Fill Attributes

VOID vqf_attributes (handle, attrib)
WORD handle
WORD attrib [5]

ČIŠĆENJE RADNE STANICE
Code: 03 Clear Workstation

VOID v_clrwk(handle)
WORD handle

Čišćenje radne stanice je hardverski zavisan pojam. Ako je u pitanju ekran, biće popunjen bojom pozadine. Isto će se desiti i sa kamerom. Kod crtača će jednostavno biti uništen bafer i postat signal za stavljanje novog papira a isto će se desiti i za štampač. Čišćenje Metafile uređaja izvodi se samo upisivanjem odgovarajućeg koda u fajl na disku.

Otvaranjem fizičke radne stanice vrši se i njeno čišćenje, dok se pri otvaranju virtualne radne stanice ništa ne događa.

AŽURIRANJE RADNE STANICE
Code: 04 Update Workstation

VOID v_updwk(handle)
WORD handle

Za spore izlazne uređaje (štampač i crtač) otvaraju se posebni baferi u koje se upisuje sve ono što treba poslati uređaju. Sadržaj bafera će biti poslat uređaju tek pozivom ove funkcije. U slučaju štampača to je niz tačaka organizovanih u linije a u slučaju crteža to je niz komandi u HPGL u ili nekom drugom komandnom grafičkom jeziku.

UČITAVANJE FONTOVA
Code: 77 Load Fonts

WORD vst_load_fonts(handle, select)
WORD handle
WORD select

Učitava sve eksterne fontove za uređaj čija radna stanica ima identifikacioni broj **handle**. Parametar **select** se za sada ne koristi i treba da bude 0.

Svaki uređaj, generalno govoreći, ima drugačije definisane fontove. U fajlu ASSIGN.SYS nalaze se spiskovi raspoloživih fontova za svaki uređaj posebno. To su takozvani eksterni fontovi. Interne fontove uređaja GEM obično ne može da iskoristi. Sve to, naravno, zavisi od drajvera. Ukoliko on ume da ih iskoristi, nema nikakvih problema.

UNIŠTAVANJE FONTOVA
Code: 78 Unload Fonts

VOID vst_unload_fonts(handle, select)
WORD handle
WORD select

Uništava eksterne fontove učitane prethodnom funkcijom. Ako više virtualnih radnih stanica koristi iste fontove oni neće biti uništeni sve dok ih se sve stanice ne „odreknu". Uništavanje eksternih fontova može oslobodititi mnogo memorije, naročito kad se uzme u obzir da se svi uvek učitavaju u „paketu". U nekoj novijoj verziji bi parametar **select** trebao obezbediti mogućnost biranja fonta koji će se učitati ili uništiti. Za sada je ovaj parametar 0.

POSTAVLJANJE OBREZUJUĆEG PRAVOUGLA
Code: 81 Set Clipping Rectangle

VOID vs_clip(handle, clip_flag, pxyarray)
WORD handle
WORD clip_flag
WORD pxyarray[]

Postavlja obrezujući pravougaonik koji je određen koordinatama sa suprotne dijagonalne tačke. Ukoliko je parametr **clip_flag** jednak 0 obrezivanje se neće vršiti a ukoliko je **1** vršiće se. Sam proces obrezivanja vrši se pri izdavanju bilo kojeg grafičkog elementa i to tako da se crtaju samo one tačke čije se koordinate nalaze unutar zadatog pravougaonika. Time se vrši odbacivanje svih delova koji prelaze okvir zadatog pravougaonika, tj. kao da je neko obrezao veliku fotografiju i iz nje izvukao jedan određen izrez. Ovom funkcijom se štiti okolina prozora tako da pri ispisivanju nečega u jednom prozoru ništa drugo ne bude pokvareno. Može se slobodno reći da bi bez ove funkcije bilo praktično nemoguće raditi s prozorima.

Niz **pxyarray[]** ima četiri elementa koji određuju koordinate gornjeg levog i donjeg desnog ugla pravougaonika i to po sledećem rasporedu

(**x1,y1**) – (pxyarray[0] pxyarray [1])
(**x2,y2**) – (pxyarray[2] pxyarray [3])

Funkcije izdavanja
(OUTPUT functions)

U ovoj kategoriji se nalazi sedamnaest funkcija kojima se izdaju osnovni grafički elementi tj. tekst, izlomljene linije, popunjeni geometrijski likovi (krugovi, elipse, lukovi, isečci, zaobljen pravougaonici) itd.

Deset funkcija ove kategorije čini **GDP-grupu** (Generalized Device Primitives) tj. grupu osnovnih geometrijskih likova. Pri otvaranju fizičke radne stanice dobijaju se i podaci o tome koje su funkcije ove grupe raspoložive, no za standardnu upotrebu možete uvek smatrati da su sve raspoložive. Samo neki specijalni izlazni uređaji postavljaju ograničenje a sa takvima se verovatno nikada nećete sresti. Sve funkcije ove grupe imaju isti kod **0B** a razlikuju se samo po pomoćnom kodu [0 A]. Samo poslednja od ovih funkcija, ona koja izdaje poravnati tekst, izložena je većim realnicijama od strane izlaznih uređaja (ne može se uvek pretupiti svakoj lački kao na ekranu).

Na rezultat poziva funkcija ove kategorije u mnogome utiču atributske funkcije kao i već objašnjena „ **clip** funkcija.

IZLOMLJENA LINIJA
Code: 06 Output Polline

VOID v_pline(handle, count, pxyarray)
WORD handle
WORD count
WORD pxyarray[]

Izdaje izlomljenu liniju na radnoj stanici čiji je **handle** predat kao parametar. Izlomljena linija, koja se u originalu

naziva mnogolinija (po analogiji sa mnogouglom) je dati na sa **count** tačaka smeštenih u niz **pxyarray [2* count]**. Pri tome su **x** koordinate smeštene u elemente sa parnim (0 2 ...) a **y** koordinate u elemente sa neparnim (1 3 ...) indeksima. Niz **pxyarray[]** se može dakle definisati i kao **pxyarray [count] [2]** i tada će se koordinate tačaka zadavati kao
(x y)i=(pxyarray[i][0], pxyarray [i][1])

Ova je funkcija u principu dobro zamišljena jer se jednim pozivom može nacrtati čitava izlomljena linija (sa maksimalno 128 tačaka — za ekransku radnu stanicu). Nedostatak je međutim u tome što se ovom funkcijom mora crtati i tačka (count=1) i obična linija (count=2) a tada dolazi do velikog usporenja u radu. Zbog toga sve novije verzije kompajlera daju mogućnost pozivanja primitivnih grafičkih rutina niskog nivoa koje čine posebnu grupu takozvanu **A-liniju**

IZDAVANJE GRUPE MARKERA
Code: 07 Output Polymarker

```
VOID v_pmarker (handle, count pxyarray)
WORD handle
WORD count
WORD pxyarray [ ]
```

Parametri ove funkcije imaju praktično identično značenje kao i parametri prethodne. Jedina razlika između ove dve funkcija je u tome što **v_pline** povlači linije među zadatim tačkama a **v_pmarker** ih ne povlači već u svakoj zadatoj tački crta po jedan marker.

Marker može biti tačka, krst, kvadrat, trougao ili bilo koji grafički simbol za označavanje tačke. Oblik markera se određuje atributskom funkcijom **vsm_type**. Pri crtanju grafika funkcija markeri se koriste za označavanje bitnih tačaka na grafiku i tada se obično bira oblik krsta.

IZDAVANJE TEKSTA
Code: 08 Output Text

```
VOID v_gtext (handle x y string)
WORD handle
WORD x y
CHAR string [ ]
```

Izdaje tekst na zadatom uređaju ali ne kao niz kodova znakova već kao sliku. Ova funkcija koristi vrednosti 6 različitih atributa radne stanice kao i sistemski ili aktuelni font i na osnovu toga generiše traku visine jednog karaktera i tekućeg fonta i dužine koja zavisi od broja karaktera. Ova kva traka se potom postavlja na odgovarajuću poziciju. Ukoliko se sve ovo radi sa crtačem kao fizičkim uređajem onda veći deo ovog posla preuzima njegova interna logika kojoj se daju samo karakteri i podaci o atributima. Slično se događa i sa Metafile uređajem.

Parametri **x,y** zadaju koordinate od kojih će početi izdavanje teksta tj. mesto na koje će biti postavljena traka s tekstom. Dužina teksta koji se predaje u varijabli **string** ograničena je dužinom **intin** niza.

POPUNJENI POLIGON
Code: 09 Output Filled Area

```
VOID v_fillarea (handle count pxyarray)
WORD handle
WORD count
WORD pxyarray [ ]
```

Izdaje poligon (trougao, četvorougao, petougao itd.) popunjen tekućim uzorkom i tekućom bojom. Poligon je zadat sa **count** tačaka čije su koordinate zadate nizom **pxyarray [2*count]** po istom formatu kao za crtanje izlomljene linije, dakle **x** koordinata su na parnim a **y** koordinate na neparnim adresama. Sve su koordinate naravno u koordinatnom sistemu koji je odabran pri otvaranju radne stanice (NDC ili RC).

Uzorak kojim se poligon popunjava, njegova boja, odnos prema pozadini, oblik ivice itd. zadaju se atributskim funkcijama.

MOZAIK OG ĆELIJA
Code: 0A Output Cell Array

```
VOID v_cellarray (handle pxyarray
                  row_length el_used
                  num_rows wrt_mode
                  colarray)

        WORD handle
        WORD pxyarray [ ]
        WORD row_length el_used num_rows
        WORD wrt_mode
        WORD *colarray
```

Ovo je prilično složena i prilično nepotrebna funkcija koja se može iskoristiti za neka specijalna naučne aplikacije (crtanje fraktala i ćelijskih automata).

Mozaik se izdaje na zadatoj radnoj stanici u pravougaoniku koji je određen elementima niza **pxyarray [4]** i to tako što su zadate koordinate gornjeg levog i donjeg desnog temena.

(x1, y1) = (pxyarray [0] pxyarray [1])
(x2, y2) (pxyarray [2] pxyarray [3])

Ovaj se pravougaonik deli na **num_rows** redova i **el_used** kolona. Tako dobijenim ćelijama dodeljuju se boje koje su zapisane u nizu **colarray** koji može biti dimenzioniran kao jednodimenzionalan **[row_length* num_rows]** ili kao dvodimenzionalan **[row_length] [num_rows]**. Parametar **el_used** ne sme biti veći od **row_length**. Ukoliko je manji onda se iz svakog reda niza **colarray** uzime samo onoliko elemenata koliko je potrebno a ostali se zanemaruju. Ukupan broj ćelija ne može biti veći od dimenzije **intin** niza što znači da njegovu dimenziju treba povećati ako imate nameru da radite sa više od 128 ćelija. Pri određivanju broja ćelija treba paziti i na to da ćelije mora biti velika barem koliko i tačka na izabranom uređaju.

Ova bi se funkcija mogla iskoristiti za pogon štampača sa trakama u boji i to tako što bi ćelija bila velika 4 tačke

sistemskog softvera stoga troše mnogo vremena u optimiziranju takvih rutina dok autori aplikacija obično nemaju ni vremena ni znanja za tako nešto. Rutine koje se „kače" na ovaj vektor vrlo lako prekorače svoje dozvoljeno vreme što sistem odvodi pravo u blokadu. Preskaliranje pokreta miša je mnogo elegantnije izvesti slanjem poruke procesoru tastature koji je za njega „nadležan" i vrlo lako prihvata vrlo veliki opseg povećanja i smanjenja koraka.

Mašinska rutina mora na početku spremiti sva registre. Rutina se poziva u režimu onemogućenog prekida i takvo stanje mora ostati sve vreme njenog izvršavanja pa i po njenom završetku. Stoga se rutina mora završavati sa RTS.

CRTANJE GRAFIČKOG KURSORA
Code: 7F Exchange Cursor Draw Vector

```
VOID vex_curv (handle pusrcode, psavcode)
        WORD handle
        WORD (*pusrcode) ()
        WORD (**psavcode) ()
```

Postavlja adresu mašinske rutine zadatu parametrom **pusrcode**, u vektor preko koga se poziva rutina za posluživanje crtanja grafičkog kursora. Preciznije rečeno ova rutina se aktivira svaki put kad VDI želi da ponovo nacrta grafički kursor koji je dodeljen lokatoru a to je obično miš. Adresa koja je ranije bila u tom vektoru vraća se preko parametra ***psavcode**.

Promenu ovog vektora koriste programi koji žele koristiti grafičke kursore nestandardnog formata.

Mašinska rutina mora na početku spremiti sve registre. Rutina se poziva u režimu onemogućenog prekida i takvo stanje mora ostati sva vreme njenog izvršavanja pa i po njenom završetku. Stoga se rutina mora završavati sa RTS.

Mešovite funkcije unosa

U ovoj poslednjoj grupi su funkcije za kontrolu grafičkog kursora (prikazivanja, gašenja, promena oblika) te za očitavanje stanja miša i tastature.

OBLIK GRAFIČKOG KURSORA
Code: 6F Set Mouse Form

```
VOID vsc_form (handle pcur_form)
        WORD handle
        WORD pcur_form[37]
```

Postavlja novi oblik grafičkog kursora koji se često poistovećuje s mišem a u stvari je dodeljen bilo kom lokatoru tj. uređaju koji može zadavati poziciju tačke kao par koordinata.

Kursor se definiše kao rasterski kvadrat od 16 × 16 tačaka. Definišu se u stvari takva kvadrata. Jedan predstavlja masku a drugi sliku kursora. Oba se zadaju kao niz od 16 bitmapiranih reči (reč = 2 bajta = 16 bita). Pri iscrtavanju kursora prvo se iscrtava **maska** pa preko nje **slika** kursora. I za masku i za sliku važi da se u zadatoj boji iscrtavaju samo tačke za koje odgovarajući bit ima vrednost 1. Obično se maska iscrtava u boji pozadine a stvara se tako da bude svuda za jednu tačku šira od slike tako da oko nje obrazuje konturu. Kako se slika obično crta u nekoj od onih boja kojima se crtaju i svi ostali grafički likovi, proširena maska obezbeđuje da se kursor vidi na svim podlogama (na onoj čija je boja jednaka boji slike kursora vidi se kontura).

Svi podaci o novom grafičkom kursoru dati su parametrom **pcur_form[37]** tj. nizom od 37 brojeva. Značenje elemenata ovog niza dato je sledećom tabelom

element	funkcija
0 1	**(x,y) — vruća tačka**
2	rezervisano — mora biti 1
3	indeks boja **maske**
4	indeks boja **slike**
5 20	raster **maske**
21 36	raster **slike**

Gde je **vruća tačka** ona tačka čije koordinate se vraćaju ili zadaju kao koordinate lokatora pa samim tim i grafičkog kursora. Brojevi **x,y** daju ofset vruće tačke u odnosu na gornji levi ugao kvadrata kojim se grafički kursor definiše.

PRIKAZIVANJE KURSORA
Code: 7A Show Cursor

```
VOID v_show_c (handle reset)
```

Prikazuje kursor koji je bio „skriven" tj. ugašen funkcijom **v_hide_c**. Ukoliko je parametar **reset=0, v_show_c"** se mora pozvati isto toliko puta koliko puta je **v_hide_c** bio pozvan da bi se kursor pojavio na ekranu. To koriste neki programi da bi ostavili grafički kursor u stanju identičnom onom u kom je bio kad je program pozvan. Ako ne želite koristiti ovu mogućnost postavite u **reset** nenultu vrednost (zvanično je zapisano da to treba da bude 1) i sve će biti u redu.

„SKRIVANJE" KURSORA
Code: 7B Hide Cursor

```
VOID v_hide_c (handle)
```

Gasi grafički kursor. Kad je grafički kursor ugašen korisnik ne može koristiti lokator za bilo kakve operacije. Ova funkcija se dosta retko koristi ali se desi da neki nekorektno napisan program pri „bacanju bombi" isključi svaku mogućnost korišćenja miša pa se mašina mora resetovati.

STANJE MIŠA
Code: 7C Sample Mouse State

```
VOID vq_mouse (handle pstatus x y)
        WORD handle
        WORD *pstatus
        WORD *x *y
```

Ispituje trenutno stanje miša. Funkcija vraća podatak u položaju miša (*x *y) te o tome koja je dugme pritisnuto

Značenja svih ostalih parametara je isto kao i kod **vrq_locator** funkcije.

UNOS PROCENJIVAČEM
Code: 1D Input Valuator

```
VOID vsm_valuator (handle val_in
          val_out term status)
     WORD handle
     WORD val_in *val_out
     WORD  term
     WORD *status
```

Unosi trenutnu vrednost zadatu procenjivačem. Dodatni parametar ***status** daje informaciju o tome šta se dogodilo između dva očitavanja. Ako je vrednost ovog parametra **0**, onda se ništa nije dogodilo, ako je **1** promenjena je vrednost, a ako je **2** pritisnuta je završna tipka.

Značenja svih ostalih parametara je isto kao i kod **vrq_locator** funkcije.

UNOS IZBORA
Code: 1E Input Choice

```
WORD vsm_choice (handle, choice)
     WORD handle
     WORD *choice
```

Unosi trenutni izbor zadat biračem. Ukoliko je pritisnuta korektna tipka (na primer, jedan od funkcijskih tastera) vraća njen kod u ***choice** a vrednost funkcije je **1**. U suprotnom je vrednost funkcije **0** a ***choice** nema značenja jer nije pritisnuta korektna tipka.

UNOS STRINGA
Code: 1F Input String

```
WORD vsm_string (handle max_length
          echo_mode echo_xy string)
     WORD handle,
     WORD max_length
     WORD echo_mode
     WORD echo_xy[ ]
     CHAR string[ ]
```

Unosi string sa znakovnog uređaja ukoliko raspoloživ. Sve dok kodovi znakova (ili standardni kodovi tastera) stižu sa uređaja (ili iz bafera) funkcija se ponaša kao i **vrq_string** odnosno čeka na kad završetka ili na svih **max_length** kodova. Onog trenutka pak kao kodovi prestanu da stižu funkcija odmah vraća kontrolu programu iz eventualno izdavanje unetog stringa na ekran.

Promena mašinskih rutina

Sledeće četiri funkcije omogućavaju promenu mašinskih rutina koje poslužuju pomeranje miša, pritisak dugmeta na mišu, tajmer i iscrtavanje grafičkog kursora. Promena se vrši tako što se u odgovarajući sistemski vektor upiše adresa nove rutine, a adresa stare se vrati programu. Kako bi je kasnije mogao restaurirati.

VEKTOR TAJMERA
Code: 76 Exchange Timer Vector

```
VOID vex_timv (handle tim_addr
          otim_addr tim_conv)
     WORD handle
     WORD (*tim_addr) ()
     WORD (**otim_addr) ()
     WORD *tim_conv
```

Postavlja adresu mašinske rutine zadatu parametrom **tim_addr.** u vektor preko koga se poziva rutina za posluživanje tajmera. Adresa koja je ranije bila u tom vektoru vraća se preko parametra ***otim_addr**. Po povratku u glavni program parametar ***tim_conv** sadrži interval (u milisekundama) posle koga tajmer izaziva prekid.

Mašinska rutina mora na početku spremiti sve registre. Rutina se poziva u režimu onemogućenog prekida i takvo stanje mora ostati sve vreme njenog izvršavanja pa i po njenom završetku. Stoga se rutina mora završavati sa RTS.

VEKTOR PRITISKA DUGMETA
Code: 7D Exchange Button Change Vector

```
VOID vex_butv (handle pusrcode psavcode)
     WORD handle
     WORD (*pusrcode) ()
     WORD (**psavcode) ()
```

Postavlja adresu mašinske rutine zadatu parametrom **pusrcode**, u vektor preko koga se poziva rutina za posluživanje pritiska na dugme miša. Preciznije rečeno ova rutina se aktivira svaki put kad se promeni stanje dugmadi na mišu. Adresa koja je ranije bila u tom vektoru vraća se preko parametra ***psavcode**.

Mašinska rutina mora na početku spremiti sve registre. Rutina se poziva u režimu onemogućenog prekida i takvo stanje mora ostati sve vreme njenog izvršavanja pa i po njenom završetku. Stoga se rutina mora završavati sa RTS.

VEKTOR POMERANJA MIŠA
Code: 7E Exchange Mouse Travel Vector

```
VOID vex_motv (handle pusrcode psavcode)
     WORD handle
     WORD (*pusrcode) ()
     WORD (**psavcode) ()
```

Postavlja adresu mašinske rutine zadatu parametrom **pusrcode**, u vektor preko koga se poziva rutina za posluživanje pomeranja miša. Preciznije rečeno ova rutina se aktivira svaki put kad VDI primeti promenu položaja miša. Adresa koja je ranije bila u tom vektoru vraća se preko parametra ***psavcode.**

Neki autori koriste ovu funkciju da bi posebno uređenim rutinama vršili preskaliranje kretanja miša tj. da bi ga usporili ili ubrzali. Čak i neki proizvođači kompajlera savetuju takvu primenu no ona se u praksi pokazala kao opasna. Rad prekidnih rutina je vremenski vrlo kritičan. Autori (2*2) a svaka ćelija bi se smatrala jednom logičkom tačkom. Sve to zahteva međutim barem 4 M za iole pristojniji rezultat.

POPUNJAVANJE KONTURE
Code: 67 Contour Fill

```
VOID v_contourfill (handle x y index)
     WORD handle
     WORD x y
     WORD index
```

Popunjavanje se vrši takozvanom tehnikom prolivenog tuša. Zamislite da ortale tušem po staklenoj ploči i da se tuš proliо iz bočice. Ukoliko na naiđe na prepreku preknće čitavo staklo. Ukoliko ga pak okružite nekom konturom od čvrstog kartona preknće samo površinu unutar konture.

Prepreku ovde predstavljaju tačke boje **index**, a polazna tačka (iz koje se izliva boja) je data koordinatama **x, y**. Uzorak kojim se popunjava njegova boja odnos prema pozadini itd. zadaju se atributskim funkcijama.

POPUNJENI PRAVOUGAONIK
Code: 72 Output Filled Rectangle

```
VOID vr_recfl (handle pxyarray)
     WORD handle
     WORD pxyarray [ ]
```

Izdaje popunjeni pravougaonik koji je zadat koordinatama gornjeg levog i donjeg desnog temena

(x1, y1) – (pxyarray [0] pxyarray [1])
(x2, y2) – (pxyarray [2] pxyarray [3])

a uzorak kojim se popunjava njegova boja odnos prema pozadini itd. zadaju se atributskim funkcijama. Ivica pravougaonika se nikad ne izdaje ovom funkcijom. Ukoliko vam to treba koristite funkciju **v_bar**.

GDP-grupa

GDP grupu sačinjavaju osnovne funkcije za izdavanje jednostavnih geometrijskih likova i teksta. Svi standardni uređaji podržavaju svih deset funkcije ove grupe a ukoliko koristite neke posebne morate proveriti funkcijom **vq_extnd** ili odmah po otvaranju radne stanice spisak elemenata niza **work_out** (15 ... 24). Glavni funkcijski kod za sva ove funkcije je 0B a konkretna funkcija se bira pod-kodom.

„PRUGA"
Code: 0B—1 Output Bar

```
VOID v_bar (handle pxyarray)
     WORD handle
     WORD pxyarray [ ]
```

„Pruga" u stvari nije ništa drugo do popunjeni pravougaonik pa je i ova funkcija vrlo slična funkciji **v_recfl**. Jedini dodatak kod ove funkcije je to što se njom može izdati i pravougaonik sa ivicom čije se debljina i oblik određuju odgovarajućim atributima.

Značenje parametara je isto kao i kod **v_recfl**.

KRUŽNI LUK
Code: 0B—2

```
VOID v_arc (handle x y radius begang endang)
     WORD handle
     WORD x y
     WORD radius,
     WORD begang
     WORD endang
```

Izdaje kružni luk na zadatu radnu stanicu. Luk je određen centrom kruga **(x y)** njegovim poluprečnikom **(radius)** te početnim **(begang)** i krajnjim **(endgang)** uglom. Ugao se meri od zamišljene horizontalne linije koja polazi iz centra kruga i „ide" od desne ivice radne površine (ekrana, papira itd.) a povećava mu se vrednost u desetim delovima stepena u smeru suprotnom kretanju kazaljke na satu. Matematičari bi na sve ovo rekli da se ugao određuje u odnosu na pozitivan smer x-ose. Sve u svemu za vertikalnu liniju se daje ugao 900 što odgovara vrednosti od 90 stepeni.

Luk se crta u smeru suprotnom kretanju kazaljke na satu i to počev od **begang** pa do **endang**.

Oblik linije njena boja odnos prema pozadini itd. zavise od tekućih vrednosti atributa.

KRUŽNI ISEČAK
Code: 0B—3 Output Pieslice

```
VOID v_pieslice (handle x y radius begang endang)
     WORD handle
     WORD x y
     WORD radius
     WORD begang
     WORD endang
```

Izdaje popunjeni kružni isečak određen na isti način kao i kružni luk. Pošto je u anglosaksonskom području pita kružnog oblika najčešći kolač to se ovakav isečak tamo naziva „parče pite" a grafikon od sečaka se naziva „pita grafikon". U našim krajevima pita je češće pravougaonog oblika a prontetni svečani kolač je torta, pa naši programeri u žargonu koriste naziv „torta-dijagram" ili samo torta. Toliko o kulinarstvu (za sada).

Uzorak kojim se popunjava njegova boja odnos prema pozadini itd. određeni su tekućim vrednostima atributa.

KRUG
Code: 0B—4 Output Circle

```
VOID v_circle (handle x y radius)
     WORD handle
     WORD x y
     WORD radius
```

Izdaje popunjeni krug određen koordinatama centra **(x y)** i poluprečnikom **(radius)**. Uzorak kojim se popunjava njegova boja odnos prema pozadini itd. određeni su tekućim vrednostima atributa.

ELIPTIČNI LUK
Code: 0B—6 Output Elliptical Arc

```
VOID v_ellarc (handle  x  y  xradius
               yradius  begang  endang)
     WORD handle
     WORD x  y
     WORD xradius  yradius
     WORD begang
     WORD endang
```

Izdaje eliptični luk na zadatoj radnoj stanici. Kao i kod kružnog luka, zadaju se početni (**begang**) i krajnji (**endang**) ugao. Elipsa je određena koordinatama centra (**x, y**) i dužinama poluosa (**xradius yradius**).

Oblik linije, njena boja, odnos prema pozadini itd. zavisi od tekućih vrednosti atributa.

ELIPTIČNI ISEČAK
Code: 0B—7 Output Elliptical Pieslice

```
VOID v_ellpie (handle  x  y  xradius
               yradius  begang  endang)
     WORD handle,
     WORD x  y
     WORD xradius, yradius
     WORD begang
     WORD endang
```

Izdaje popunjeni eliptični isečak na zadatoj radnoj stanici. Isečak je zadat na isti način kao i luk. Uzorak kojim se popunjava, njegova boja, odnos prema pozadini itd. zavise od tekućih vrednosti atributa.

ELIPSA
Code: 0B—5 Output Ellipse

```
VOID v_ellipse (handle  x  y  xradius  yradius)
     WORD handle
     WORD x  y
     WORD xradius  yradius
```

Izdaje popunjenu elipsu zadatu koordinatama centra (**x,y**) i dužinama dve poluose (**xradius, yradius**). Uzorak kojim se popunjava, njegova boja, odnos prema pozadini itd., zavise od tekućih vrednosti atributa.

ZAOBLJENI PRAVOUGAO
Code: 0B—8 Output Rounded Rectangle

```
VOID v_rbox (handle  pxyarray)
     WORD handle
     WORD pxyarray []
```

Izdaje pravougaonik sa zaobljenim temenima, koji je zadat koordinatama gornjeg levog i donjeg desnog temena

(x1, y1) – (pxyarray [0], pxyarray [1])
(x2, y2) – (pxyarray [2], pxyarray [3])

a oblik linije, njena boja, odnos prema pozadini itd. zavise od tekućih vrednosti atributa.

POPUNJENI ZAOBLJENI PRAVOUGAO
Code: 0B—9 Output Filled Rounded Rectangle

```
VOID v_rfbox (handle  pxyarray)
     WORD handle
     WORD pxyarray [ ]
```

Izdaje popunjeni pravougaonik zaobljenih temena. Pravougaonik je zadat na isti način kao i u prethodnoj funkciji, a uzorak kojim se popunjava, njegova boja, odnos prema pozadini itd. zavise od tekućih atributa.

IZDAVANJE PORAVNATOG TEKSTA
Code: 0B—A Output Justified Text

```
VOID v_justified (handle  x  y  string  length
                  word_space  char_space)
     WORD handle
     WORD x  y
     CHAR string[ ]
     WORD length
     WORD word_space
     WORD char_space
```

Izdaje poravnat tekst na zadatoj radnoj stanici. Tekst se predaje u varijabli **string**, a njegova dužina ne može biti veća od dužine niza **intin**. Tekst počinje od tačke zadatih koordinata (**x y**), a razmaci između reči i/ili slova podešavaju se tako da izdati tekst bude dugačak tačno **length** tačaka (u NDC ili RC sistemu, zavisno o tome kako je radna stanica otvorena). Ukoliko je vrednost parametra **word_space** =0, onda se neće vršiti razmicanje reči, a ako je vrednost parametra **char space** 0 onda se neće vršiti razmicanje slova.

Razmicanje slova (umetanjem tačaka) je prilično delikatna operacija koja se lako izvodi za ekran, ali za običan iglični štampač može predstavljati pravu moru pošto za takvu „egzibiciju" štampač mora obavezno preći u grafički mod. Stoga treba maksimalno izbegavati korišćenje ove funkcije samo za rad sa tekstom na štampaču. Ukoliko pak radite crtež u kome se pojavljuje i tekst, onda nema nikakvih problema pošto je tad štampač i onako u grafičkom modu i izdaje čitavu sliku odjednom.

Oblik, boja, položaj, visina itd. izdatog teksta zavise od tekućih atributa.

Atributske funkcije

(Attribute Functions)

U ovoj kategoriji se nalaze 22 funkcije koje se mogu podeliti u pet grupa, zavisno od toga koje atribute postavljaju. U prvoj **opštoj grupi** su dve funkcije koje postavljaju opšte atribute, tj. mod pisanja i paletu boja. Druga grupa se pet funkcija određuje **atribute linije** (tip, debljinu, boju, način završavanja), a treća sa in funkcija određuje **atribute markera** (tip, visinu, boju). Četvrta grupa ima sedam funkcija za određivanje **atributa teksta** (visinu, boju, font, ugao, fino podešavanje, specijalne efekte), a peta ima pet

Funkcije unošenja ZAHTEVOM

(Request Input Functions)

UNOS LOKATOROM
Code: 1C Input Locator

```
VOID vrq_locator (handle, x  y  xout
                  yout  term)
     WORD handle
     WORD x  y
     WORD *xout  *yout
     WORD *term
```

Unosi poziciju tačke pomoću lokatora. Po pozivu ove funkcije grafički kursor se pojavljuje na ekranu u tački zadatoj parametrima **x y**. Potom ga korisnik slobodno pomera lokatorom i kad se odluči za neku tačku pritisne završnu tipku. Potom se koordinate izabrane tačke vraćaju u parametrima ***xout, *yout**, a kod pritisnute tipke u ***term**. Ukoliko se kao lokator koristi kursorski tasteri, završna tipka je bilo koji drugi taster čiji se ASCII kod vraća u ***term**. Ukoliko je miš lokator završne tipke su dugmad miša a vraćeni kod je **0×20** za prvo, **0×21** za drugo dugme itd. (VDI podrzava i miševe sa više dugmadi). Grafički kursor će neposredno pre vraćanja kontrole programu biti vraćen u stanje u kom je bio pre pozivanja ove funkcije.

UNOS PROCENJIVAČEM
Code: 1D Input Valuator

```
VOID vrq_valuator (handle  valuator_in
                   valuator out  terminator)
     WORD handle
     WORD valuator_in
     WORD *valuator_out
     WORD *terminator
```

Unosi vrednost iz opsega [1 – 100] pomoću procenjivača. Po pozivu ove funkcije, uređaj za procenjivanje se postavi u stanje koje odgovara vrednosti parametra **valuator_in**. Potom korisnik može menjati stanje procenjivača sve dok ne pritisne završnu tipku, kada se rezultat izabranog stanja vraća kao vrednost parametra ***valuator out**, a kod završne tipke kao vrednost parametra ***terminator**.

Ako se kao procenjivač koriste gornji i donji kursorski taster, svaki pritisak povećava ili smanjuje vrednost za 10 ili za 1 ako je shift taster pritisnut. Završna tipka je u tom slučaju bilo koji drugi taster čiji se ASCII kod vraća. Za druge uređaje završni kod se dobija kao **0×20+ord**, gde je **ord** redni broj dugmeta na uređaju (prvo dugme ima redni broj 0). Sve to naravno ovisi o drajveru.

UNOS IZBORA
Code: 1E Input Choice

```
VOID vrq_choice (handle  ch_in  ch_out)
     WORD handle
     WORD ch_in
     WORD *ch_out
```

Unosi broj tipke izabrane na uređaju za biranje. Ukoliko je izbor korektan (o čemu odlučuje drajver), vraća se redni broj tipke (1, 2, ...) u parametru ***ch_out**, a ukoliko nije, vraća se u istom parametru vrednost parametra **ch_in** koji predstavlja inicijalnu i podrazumevanu vrednost. Ta vrednost može biti i broj van opsega rednih brojeva tipki i tada jednostavno označava grešku. Funkcija čeka da korisnik pritisne neku tipku i neće vratiti kontrolu programu sve dok se to ne dogodi. Kao uređaj za biranje se obično koristi skup funkcijskih tastera.

UNOS STRINGA
Code: 1F Input String

```
VOID vrq_string (handle  max_length
                 echo_mode  echo_xy  string)
     WORD handle
     WORD max_length
     WORD echo_mode
     WORD echo_xy[ ]
     CHAR string [ ]
```

Unosi string sa uređaja za unos znakova. Kodovi znakova (obično se radi sa ASCII kodovima) se akumuliraju sve dok se ne skupi **max_len** kodova ili ne stigne završni kod (obično je to CR). Tada se string vraća programu u parametru **string**. Ukoliko je **echo_mode**=0, tu je i kraj izvršavanja funkcije. U suprotnom se uneti string izdaje na ekranu shodno tekućim vrednostima atributa teksta, a na poziciji zadatoj nizom **echo_xy [2]** čiji elementi daju koordinate pozicije

(x,y) – (echo_xy[0], echo_xy[1])

Ako je parametar **max_len negativan**, uzima se njegova apsolutna vrednost, a akumuliraju se standardni kodovi tastera.

Funkcije unošenja PROBOM

(Sample Input Functions)

UNOS LOKATOROM
Code: 1C Input Locator

```
WORD vsm_locator (handle  x  y  xout
                  yout  term)
     WORD handle
     WORD x  y
     WORD *xout  *yout
     WORD *term
```

Unosi trenutnu poziciju zadatu lokatorom. Ne prikazuje grafički kursor na ekranu. Ovo je prava funkcija a vrednost koju vraća daje informaciju o tome šta se desilo sa lokatorom između dva očitavanja. Vrednost 0 znači da se ništa nije dogodilo, 1 znači da su koordinate promenjene ali da nije pritisnuta završna tipka, 2 znači da je pritisnuta završna tipka ali da koordinate nisu promenjene i konačno 3 znači da su i koordinate promenjene i završna tipka pritisnuta.

Funkcije unosa

(Input Fonctions)

Ova kategorija sadrži 18 funkcija od kojih je 9 namenjeno radu sa **logičkim uređajima unosa**. Četiri funkcije se koriste za promenu mašinskih rutina za opsluživanje miša, grafičkog kursora i tajmera, a preostalih pet za direktnu kontrolu grafičkog kursora i očitavanje stanja tastature i miša.

GEM VDI rešava unos podataka modelom **logičkih uređaja unosa** koji teorijski mogu postojati na svakoj radnoj stanici, iako je praktično teško zamisliti kako bi se štampačem unosila pozicija grafičkog kursora, na primer. Za razliku od TOS-a i MS-DOS-a koji poznaju uređaj nazvan konzola, sastavljen od tastature i displeja, VDI sma tra ekran uređajem, otvara njegovu radnu stanicu i dodeljuje joj logičke uređaje unosa koji koriste delove tastature ali i miša. Treba dakle uvek paziti da se ne koriste i TOS odnosno MS-DOS funkcije i VDI funkcije za isti posao jer to može stvoriti veliku zbrku kako na ekranu tako i u samom programu.

GEM VDI poznaje četiri logička uređaja unosa.

LOKATOR (locator)

ili indikator pozicije je uređaj kojim se unosi pozicija grafičkog kursora u važećem koordinatnom sistemu (**NDC** ili **RC**). U tu svrhu se obično koristi **miš**, **svetlosno pero** ili **grafička tabla**, a mogu se koristiti i **kursorski tasteri**.

PROCENJIVAČ (valuator)

je uređaj za unošenje skalarnih podataka. Obično je to kombinacija potenciometra i A/D konvertora ili je u pitanju kakav brojčanik čijim se okretanjem određuje vrednost koju treba uneti. U nedostatku ovakvih uređaja koriste se **dva kursorska** tastera čiji pritisak povećava ili smanjuje vrednost za deset ili za jedan (u kombinaciji sa shift tasterom). Opseg dozvoljenih vrednosti je [1 ... 100] što je za sadašnje fizičke ulazno-izlazne jedinice sasvim dovoljno. Već u sledećoj generaciji će to biti nedovoljno, no nema nikakvih problema da se opseg proširi na [0 ... 65536].

BIRAČ (choice device)

je uređaj sa tačno određenim (numerisanim) diskretnim stanjima od kojih se uvek bira jedno i njen redni broj se prosleđuje kao ulaz. Birač se obično realizuje kao niz prekidača. Za **ekran** kao uređaj, birač čine **funkcijski tasteri**.

ZNAKOVNI UREĐAJ (string device)

je uređaj kojim se unose niske znakova poznatije u domaćem programerskom žargonu pod nazivom **stringovi karaktera**. Za ovo se gotovo uvek koristi tastatura kao uređaj pridružen ekranskoj radnoj stanici. Nije međutim isključena mogućnost korišćenja skanera sa softverom za prepoznavanje teksta koji se može smatrati ulaznim uređajem, kamere pa čak i štampača.

NAČIN UNOŠENJA
Code: 21 Set Input Mode

```
VOID vsin_mode(handle dev_type mode)
        WORD handle
        WORD dev_type
        WORD mode
```

Određuje način unošenja podataka za zadati logički uređaj unosa na zadatoj radnoj stanici. Kod logičkog uređaja unosa se zadaje parametrom **dev type** i može imati sledeće vrednosti:

kod	uređaj
1	lokator
2	procenjivač
3	birač
4	znakovni uređaj

Način unošenja se određuje parametrom **mode** koji može imati vrednost 1 ili 2, čime se određuje unošenje **zahtevom** ili **probom**.

UNOS ZAHTEVOM

Kod unošenja zahtevom (**Request Input**) VDI aktivira odgovarajući logički uređaj unosa, uključi eventualne pokazivače (strelica miša itd.) i omogućava korisniku da mu menja stanje (pomera miša, obrće potenciometar, pritiska tastere itd.) sve dok ne pritisne određeni taster (ili dugme na mišu) koji je proglašen za **završni**. Tada se programu koji je tražio unošenje vraća **poslednje stanje** logičkog uređaja.

UNOS PROBOM

Kod unošenja probom (**Sample Input**), logički uređaj unosa se aktivira samo toliko dugo koliko je potrebno da se očita trenutno stanje koje se odmah vraće programu. Budući da se svim uređajima unosa služe ljudi čije je vreme reakcije mnogo duže od vremena izvršavanja kompjuterskih instrukcija, unošenje probom omogućava programu da uporedo sa unošenjem obnavlja još neki posao. Ovde se međutim javlja par problema. Pre svega pojam aktiviranja uređaja unosa nije sasvim precizan. Svaki pritisak tastera ili pokret miša šalju podatke u bafer koji održava TOS pa se postavlja pitanje šta će se i odakle učitati. VDI naime mora na nekom nivou da koristi usluge operativnog sistema ili bi na svakoj sledećoj verziji mašine „padao". Osim toga kad dve ili više virtualnih radnih stanica dele jednu fizičku, drajver nema mogućnost da odredi kojoj od virtualnih stanica pripada „proba". Sve to umnogome smanjuje upotrebljivost ovog načina unošenja podataka.

Četiri funkcije za unošenje podataka (svaka za po jedan tip logičkog uređaja unosa) imaju iste jedinstvene kodove nezavisne od izabranog načina unošenja. Broj parametara je međutim zavisan od načina unošenja pa se zbog toga u svim programskim jezicima uvode **dve funkcije po logičkom uređaju unosa**. Pozivom neke od tih funkcija **ne vrši se izbor načina unošenja** već se jednostavno zadovoljava forma prenosa parametara. Da ne bi bilo zabune biće odvojeno prikazane četiri funkcije za unošenje zahtevom i četiri za unošenje probom.

funkcija za određivanje **atributa popunjavanja** (tip, stil, boja, rub) ili atributa uzorka kojim se popunjava.

Sve ove funkcije bitno utiču na izvršavanje **funkcija iscrtavanja**, pa treba u radu s njima biti vrlo pažljiv. Svaki put kad program treba da nacrta novi geometrijski lik, mora prvo da postavi potrebne atribute kako se ne bi dogodilo da neki od ranije korišćenih atributa „zaluta" u novu sliku.

Opšti atributi

NAČIN ISPISIVANJA
Code: 20 Set Writing Mode

```
WORD vswr_mode (handle mode)
        WORD handle
        WORD mode
```

VDI poznaje četiri načina ispisivanja grafičkih elemenata: način zamene, providni način, XOR način i obratni providni način. Kod načina ispisivanja se zadaje parametrom **mode** a funkcija vraća isti taj kod ili kod greške. Raspored kodova dat je sledećom tabelom (koja se obično nalazi u nekom .H fajlu kao deo #define deklaracija):

MD_REPLACE	1
MD_TRANS	2
MD_XOR	3
MD_ERASE	4

Da bi se objasnili načini ispisivanja treba uvesti nekoliko oznaka. Neka je **obj** bit iz uzorka za lik koji se ispisuje (linija, tačka koje čine karaktere teksta itd.), **for** boja kojom se ispisuje (od foregound — prednji plan), **back** boja pozadine (od background — pozadina), **old** boja tačke koja je već ispisana na mestu za koje se traži nova boja tačke i **new** boja koja će se dodeliti tački. Tačke lika koji se iscrtava biće nazvane crnom ako je **obj** 1, a belom ako je **obj** 0.

MD_REPLACE način zamene

Svaka crna tačka objekta se jednostavno ispisuje u **for** boji, a svaka bela u **back** boji. Logička zavisnost je

new = (obj AND for) OR (NOT obj AND back)

MD_TRANS providni način

Samo crne tačke se ispisuju u **for** boji, dok se bele tačke ostavljaju u **old** boji. Logička zavisnost je

new = (obj AND for) OR (NOT obj AND old)

MD_XOR XOR način

Ispisivanje crnih tačaka se svodi na invertovanje boje postojećih, dok ispisivanje belih tačaka ništa ne menja. Ovaj način je u stvari direktna realizacija XOR logičke funkcije new = obj XOR old

MD_ERASE obratni providni način

Ovaj se način još zove **brišući** jer se ispisivanje crnih tačaka vrši u **old** boji, što će reći da se uopšte ne ispisuju. Tako će pravougaonik popunjen crnim tačkama pri ispisivanju u ovom modu jednostavno nestati. Bele tačke se pak ispisuju u **for** boji. Logička zavisnost je

new = (obj AND old)OR(NOT obj AND for)

Logička operacija između bita uzorka **obj** i boje izvodi se tako što se izvodi operacija sa svakom bit ravni posebno. Može se dakle smatrati da grafički lik u boji ima jednu ravan u kojoj je boja tačke kodirana sa **n**-bita (1, 2 ili 4 za ekran) ili da ima **n** ravni koje zajedno određuju boju tačke. Bit mapiran broj koji je dobijen spajanjem bitova sa svih bit ravni za datu tačku naziva se **vrednost tačke** i predstavlja njenu boju. Može se dakle precizno reći da su **old**, **for**, **back**, **new** vrednosti tačaka za atributskim funkcijama zadate boje. O razlici između vrednosti tačke i indeksa boje biće više reči kod **rasterskih funkcija**, gde će se sve to posmatrati na „mikro" nivou ili na nivou skupa tačaka koje čine raster slike.

Zadavanje načina ispisivanja utiče na sve funkcije iz kategorije **funkcije iscrtavanja** sem funkcije **v_cellarray** koja je i bez toga dovoljno komplikovana.

POSTAVLJANJE PALETE
Code: 0E Set Colour Representation

```
VOID vs_color (handle index rgb_in)
        WORD handle
        WORD index
        WORD rgb_in[3],
```

Postavlja fizičku vrednost boje u paleti. Veličina palete zavisi od konkretnog uređaja i može se naći u **work_out** 13 po otvaranju radne stanice ili pozivu **vq_extnd** funkcije.

Fizička vrednost boje je uređena trojka (**R G B**) gde brojevi **R G** i **B** daju intenzitet crvene, zelene i plave hromatske komponente čijim se mešanjem dobija boja. Na taj način rade praktično svi uređaji koji mogu zapisati i/ili reprodukovati informaciju o boji.

VDI dozvoljava da svaka komponenta dobije vrednost **[0 1000]**, što znači da možete zadati 10^9 (i slovima devet na deveti) ili tačno milijardu različitih boja i nijansi. Druga je stvar da li će konkretni uređaj moći da ih reprodukuje.

Parametar **index** daje redni broj boje u paleti. Indeksi 0 i 1 uvek postoje i indeks 0 je uvek boja pozadine. Fizička vrednost boje je zadata elementima niza **rgb_in** po rasporedu

rgb_in[0] R
rgb_in[1] G
rgb_in[2] B

Boje postavljene ovom funkcijom važe za sva ispisivanja koja se vrše VDI funkcijama. Sve ostale funkcije koje određuju boje odrađuju u stvari samo indekse.

Atributi linije

BOJA LINIJE
Code: 11 Set Line Colour

```
WORD vsl_color(handle,color_index)
     WORD handle
     WORD color_index
```

Postavlja boju za crtanje linija. Uvek postoje barem dve boje (0 i 1). Postavljena boja ostaje važeća za crtanje svih linija i to sve dok istom ovom funkcijom ne bude promenjena. Funkcija vraća vrednost postavljenog koda ili kod greške.

TIP LINIJE
Code: 0F Set Line Type

```
WORD vsl_type(handle,style)
     WORD handle
     WORD style
```

Postavlja tip linije. Broj tipova linije zavisi od uređaja, no svi uređaji treba da podržavaju sedam osnovnih tipova. Parametar **style** daje kod tipa linije. Raspored kodova dat je sledećom tabelom (koja se obično nalazi u nekom .H fajlu kao deo #define deklaracija). U komentarima je dat bit uzorak linije onako kako je zapisan u sistemu — kao 16 bitova jedne reči.

tip	kod	uzorak
SOLID	1	1111111111111111
LONGDASH	2	1111111111110000
DOT	3	1110000011100000
DASHDOT	4	1111111000111000
DASH	5	1111111100000000
DASH2DOT	6	1111000110011000
USERLINE	7	korisnički definisan

Funkcija vraća vrednost postavljenog koda ili kod greške.

KORISNIČKI TIP LINIJE
Code: 71 Set User Defined Line Style

```
VOID vsl_udsty(handle,pattern)
     WORD handle
     WORD pattern
```

Ovom se funkcijom zadaje uzorak korisničku definisanog tipa linije (kod=7). Parametar **pattern** sadrži bit-mapu uzorka analognu bit mapama datim za standardni set tipova. Svaki bit čija je vrednost 1 predstavlja crnu a svaki čija je vrednost 0 belu tačku u smislu koji je naveden kod funkcija **vswr_mode**.

DEBLJINA LINIJE
Code: 10 Set Line Width

```
WORD vsl_width(handle,width)
     WORD handle
     WORD width
```

Debljina linije je parametar koji u mnogome zavisi od konkretnog uređaja. Vrednost koju funkcija vraća je najbliža raspoloživa vrednost za onu koja je zadata parametrom **width**. Ova vrednost nikad nije veća od zadate. Za veće debljine se može dogoditi da linija bude nacrtana kao puna (tip SOLID, kod 1) iako je neki drugi tip zadat. Čak i za ekran nisu uvek moguće sve vrednosti. Raspoložive debljine zavise osim toga i od drajvera za konkretan uređaj.

KRAJEVI LINIJA
Code: 6C Set Line End Styles

```
VOID vsl_ends(handle,beg_style,end_style)
     WORD handle
     WORD beg_style
     WORD end_style
```

Krajevi linija, tj. početak i kraj mogu se definisati kao četvrtasti, streličasti i zaobljeni. Strelica na početku ili završetku linije je vrlo pogodna za izradu tehničkih crteža. Parametrima **beg_style** i **end_style** se zadaju kodovi početka i završetka linije. Kao i svi drugi atributi i ovi ostaju aktivni sve dok se ne promene pa treba biti pažljiv kako ne bi dobili mnogo strelica tamo gde vam ne trebaju. Raspored kodova dat je sledećom tabelom (koja se obično nalazi u nekom .H fajlu kao deo #define deklaracija).

tip	kod
SQUARED	0
ARROWED	1
ROUNDED	3

Atributi markera

TIP MARKERA
Code: 12 Set Marker Type

```
WORD vsm_type(handle,symbol)
     WORD handle
     WORD symbol
```

Ovom funkcijom se bira tekući tip markera. Broj raspoloživih markera zavisi od uređaja no svi uređaji treba da podržavaju šest osnovnih tipova. Kod tipa se zadaje parametrom **symbol**. Standardni tipovi su:

kod	tip
1	tačka
2	krst
3	zvezdica
4	kvadrat
5	dijagonalni krst
6	romb

Funkcija vraća vrednost postavljenog koda ili kod greške.

VISINA MARKERA
Code: 13 Set Marker Height

```
WORD vsm_height(handle,height)
     WORD handle
     WORD height
```

Postavlja tekuću vrednost visina markera (u NDC ili RC sistemu, ovisno o tome kako je radna stanica otvore-

```
     WORD pxyarray[ ]
     MFDB psrcMFDB
     JFDB pdesMFDB
     WORD color_index[ ]
```

Kod prozirnog kopiranja izvorni rastler može imati samo jednu bit ravan i mora biti u memoriji dok odredišni i dalje može biti kako u memoriji tako i u displeju uređaja. Na raspolaganju su samo četiri operacije s blokovima koje vrlo približno odgovaraju načinima izdavanja grafičkih likova specificiranim pomoću funkcije **vswr_mode**. Kod operacije, odnosno načina kopiranja zadaje se parametrom **wr_mode**. Raspored kodova dat je sledećom labelom (koja se obično nalazi u nekom .H fajlu kao deo #define deklaracija). Radi se o potpuno istoj labeli kao za **vswr_mode**.

MD_REPLACE	1
MD_TRANS	2
MD_XOR	3
MD_ERASE	4

Niz **color_index 2** sadrži dva indeksa boja. Radi jednostavnosti zapisivanja biće obeležne kao

r_for -color_index 0
r_back -color_index 1

Pored toga biće korišćene i oznake uvedene za **vswr_mode**. Dakle **obj** je bit izvornog rastera i shodno njegovoj vrednosti tačka će biti nazvana crnim ako je **obj** 1 a belom ako je **obj** 0. Boja tačke odredišnog rastera je **old**, a nova boja je tačke **new**. Logička operacija između izvornog bita **obj** i boje izvoda sa tako što se izvod operacija sa svakom bit ravni posebno. Može se dakle smatrati da odredišni raster ima jednu ravan u kojoj je boja tačke kodirana sa n-bita (1,2 ili 4 za ekran) ili da ima n ravni koje zajedno određuju boju tačke.

MD_REPLACE način zamene

Svaka crna tačka bojekta se jednostavno ispisuje u **r_for** boji a svaka bela u **r_back** boji. Logička zavisnost je

new -(**obj** AND **r_for**) OR (NOT **obj** AND **r_back**)
Line Below 10 **MD_TRANS providni način**

Samo crne tačke se ispisuju u **r_for** boji dok se bele tačke ostavljaju u **old** boji. Logička zavisnost je

new -(**obj** AND **r_for**) OR (NOT **obj** AND **old**)

MD_XOR XOR način

Ispisivanje crnih tačaka se svodi na invertovanje boje postojećih dok ispisivanje belih tačaka ništa ne menja. Ovaj način je u stvari direktna realizacija XOR logičke funkcije:

new -**obj** XOR **old**

MD_ERASE obratni providni način

Ovaj se način još zove i **brisući** jer se ispisivanje crnih tačaka vrši u **old** boji što će reći da se uopšte ne ispisuju. Tako će pravougaonik popunjen crnim tačkama pri ispisivanju u ovom modu jednostavno nestati. Bele tačke se pak ispisuju u **r_back** boji. Logička zavisnost je

new -(**obj** AND **old**) OR(NOT **obj**AND **r_back**)

Svi ostali parametri imaju isto značenje kao i u **vro_cpyfm**.

NAPOMENA: Kod nekih kompajlera srešćete delimično izmenjena imena dve upravo objašnjene funkcije za kopiranje. Izmenjena imena vas mogu zbuniti jer se ime **vrp_cpyfm** pojavljuje u obe varijante ali na različitom mestu. Stoga je najbolje ova dve funkcije uvek razmatrati u paru. Može se dakle dogoditi da vaš kompajler umesto para (**vro_cpyfm**,**vrt_cpyfm**) koristi par (**vro_cpyform**,**vro_cpyfm**) što vas lako može navesti na potpuno pogrešnu stranu. Na svu sreću **vrt_cpyfm** ima **šest** parametra a **vro_cpyfm** ih ima **pet** što predstavlja potpuno pouzdan znak za identifikaciju koja funkcija nosi koje ime i šta radi.

TRANSFORMACIJA FORME
Code: 6E Transform Form

```
VOID vr_trnfm(handle,psrcMFDB,pdesMFDB)
     WORD handle
     MFDB psrcMFDB
     MFDB pdesMFDB
```

Ovom se funkcijom vrši transformisanje rastera iz **NDC** u **RC** formu i obratno. Uvek se vrši transformacija izvornog rastera (određenog sa **MFDB psrcMFDB**) u odredišni (određen sa **MFDB pdesMFDB**). Pošto obe funkcije za kopiranje rasterskih blokova zahtevaju rad u **RC** sistemu, ova funkcija je ključna za prenosivost programa jer se prenosni programi kompletno pišu u **NDC** sistemu. Zahvaljujući tome možete učitati sliku sa **PC** mašine, obraditi je i vratiti je na **PC** mašinu.

UZIMANJE TAČKE
Code: 69 Get Pixel

```
VOID v_get_pixel(handle,x,y,pel,index)
     WORD handle
     WORD x,y
     WORD pel
     WORD index
```

Za tačku čije koordinate su date parametrima **x**,**y** ova funkcija u ***pel** upisuje bit map-ran broj dobijen spajanjem bitova svih bit ravni. Za zadatu tačku taj broj nosi naziv **vrednost tačke**. Funkcija ***index** pak upisuje indeks boje. Ovaj indeks boje će prilikom crtanja ili popunjavanja proizvesti baš onaj raspored bitova po bit-ravnima koji je upisan u ***pel**. Preslikavanje vrednosti tačke i indeksa boje je zavisno od uređaja koji se koristi odnosno od njegovog drajvera.

vi slika. Ovakvi blokovi se opisuju posebnom strukturom **MFDB** (Memory Form Definition Block) koja čuva sva podatke koji su bitni za pristup i manipulisanja blokom tačaka — rasterom.

```
TYPEDEF STRUCT{

        BYTE    *fd_addr
        WORD    fd_w
        WORD    fd_h
        WORD    fd_wdwidth
        WORD    fd_stand
        WORD    fd_nplanes
        WORD    fd_r1
        WORD    fd_r2
        WORD    fd_r3
}MFDB;
```

Značenja pojedinih stavki su prilično jednostavna. **fd_addr** čuva adresu početka bloka. **fd_w** je širine a **fd_h** visina bloka u tačkama, dok je **fd_wdwidth** širina bloka u rečima tj. **fd_wdwidth = fd_w/16**. Ukoliko je **fd_addr=0**, to znači da se blok ne nalazi u memoriji računara već u displeju uređaja. To što je displej ekrana ili bafer štampača takođe deo fizičke memorije računara ne uzima se u obzir. Onog trenutka kad je otvorena odgovarajuća radna stanica, svi njeni baferi se smatraju delom uređaja. Pomeranje blokova po ekranu se najčešće i vrši navođenjem nultih adresa.

Stavka **fd_stand** je flag. Ako je 0 onda je raster u hardverski zavisnoj formi, tj. u **RC** sistemu. Ako je pak ova stavka 1 onda je raster u standardnom tj. **NDC** sistemu. Poslednja korišćena stavka je **fd_nplanes** koja daje podatak koliko bit ravni postoji, tj. koliko bita u bloku odgovara jednoj tački rastera. Preostale tri stavke se za sada ne koriste i moraju biti jednake 0.

NEPROZIRNO KOPIRANJE
Code: 6B Copy Raster Opaque

```
VOID vro_cpyfm(handle vr_mode
               pxyarray psrcMFDB pdesMFDB)
     WORD handle
     WORD vr_mode
     WORD pxyarray[ ]
     MFDB psrcMFDB
     MFDB pdesMFDB
```

Pri neprozirnom kopiranju i izvorni i odredišni blok mogu biti kako u memoriji tako i u uređaju sa proizvoljnim brojem bit ravni. Pre pozivanja ove funkcije potrebno je popuniti izvornu (**psrcMDFB**) odredišnu (**pdesMDFB**) informacionu strukturu. Parametar **pxyarray [8]** određuje izvorni i odredišni pravougaonik u **RC** sistemu (i izvorni i odredišni raster moraju biti u **RC** sistemu). Ukoliko se oba rastera odnose na isti uređaj (gotovo uvek se radi o ekranu) pravougaonici se mogu i preklapati jer funkcija uvek prepiše odredišni blok u poseban bafer pa tamo izvrši potrebne operacije. Izvorni i odredišni pravougaonik su određeni na standardni način: koordinatama gornjeg levog i donjeg desnog temena

```
{xs1.ys1} – {pxyarray [0] pxyarray [1]}
{xs2.ys2}   {pxyarray [2] pxyarray [3]}
{xd1.yd1}   {pxyarray [4] pxyarray [5]}
{xd2.yd2} — {pxyarray [6] pxyarray [7]}
```

gde se prva dva tačke {(xs1.ys1) i (xs2,ys2)} odnose na izvorni a druge dve ({xd1.yd1}) na odredišni pravougaonik.

Parametar **vr_mode** sadrži kod operacije koja se izvršava u toku kopiranja. Neka radi jednostavnosti **sour** označava izvorni sa **dest** odredišni blok a sa **new** blok koji će se iz bafera upisati umesto **dest** po završetku izvođenja operacija. Postoji 16 različitih operacija kojima se mogu dobiti vrlo zanimljivi efekti iako se najčešće koristi proste kopiranja. Raspored kodova dat je sledećom tabelom (koja se obično nalazi u nekom .H fajlu kao deo #define deklaracija)

```
ALL_WHITE      0
S_AND_D        1
S_AND_NOTD     2
S_ONLY         3
NOTS_AND_D     4
D_ONLY         5
S_XOR_D        6
S_OR_D         7
NOT_SORD       8
NOT_SXORD      9
D_INVERT      10
NOT_D         11
S_OR_NOTD     12
NOTS_OR_D     13
NOT_SANDD     14
ALL_BLACK     15
```

Značenje operacija biće dato u terminima logičkih operacija AND, OR, NOT i XOR jer tako možete odmah videti šta se dešava sa tačkama rastera.

```
ALL_WHITE     new = 0
S_AND_D       new = sourANDdest
S_AND_NOTD    new = sourAND(NOTdest)
S_ONLY        new = sour
NOTS_AND_D    new = (NOT sour)ANDdest
D_ONLY        new = dest
S_XOR_D       new = sourXORdest
S_OR_D        new = sourORdest
NOT_SORD      new = NOT(sourORdest)
NOT_SXORD     new = NOT(sourXORdest)
D_INVERT      new = NOTdest
S_OR_NOTD     new = sourOR(NOTdest)
NOT_S         new = NOTsour
NOTS_OR_D     new = (NOTsour)ORdest
NOT_SANDD     new = NOT(sourANDdest)
ALL_BLACK     new = 1
```

PROZIRNO KOPIRANJE
Code: 79 Copy Raster Transparent

```
VOID vrt_cpyfm(handle vr_mode pxyarray
               psrcMFDB pdesMFDB color_index)
     WORD handle
     WORD vr_mode
```


na). Vrednost koju funkcija vraća je najbliža raspoloživa vrednost za onu koja je zadata parametrom **height**. Ova vrednost nikad nije veća od zadate.

BOJA MARKERA
Code: 14 Set Marker Colour

```
WORD vsm_color(handle color_index)

     WORD handle
     WORD color_index
```

Postavlja boju markera. Uvek postoje barem dve boje (0 i 1). Postavljena boja ostaje aktivna za crtanje svih markera i to sve dok istom ovom funkcijom ne bude promenjena. Funkcija vraća vrednost postavljenog koda ili kod greške.

Atributi teksta

VISINA TEKSTA (apsolutna)
Code: 0C Set Text Height (Absolute)

```
VOID vst_height(handle height char_width
                char_height cell_width
                cell_height)
     WORD handle,
     WORD height
     WORD *char_width *char_height
     WORD *cell_width *cell_height
```

Postavlja tekuću visinu teksta merenu duž y-ose koordinatnog sistema (NDC ili RC ovisno o tome kako je radna stanica otvorena. Visina se zadaje parametrom **height**. Funkcija postavlja najbližu raspoloživu vrednost visine koja nikad nije veća od zadate. Četiri preostala parametra (koji su definisani kao pointeri) služe za vraćanje podataka koji važe za tekući font. Njihovo značenje vidite u sledećoj tabeli

***char_width**	širina karaktera
***char_height**	visina karaktera
***cell_width**	širina ćelije karaktera
***cell_height**	visina ćelije karaktera

Visine su ovde date kao rastojanje između osnovne i vršne linije. Ukoliko je tekući font proporcionalan onda se visine karaktera i ćelije karaktera odnose na najširi karakter u fontu.

VISINA TEKSTA (u tačkama)
Code: 6B Set Text Height (Points)

```
VOID vst_point(handle point char_width
               char_height cell_width
               cell_height)
     WORD handle
     WORD point,
     WORD *char_width *char_height,
     WORD *cell_width *cell_height
```

Sve što je navedeno za prethodnu funkciju važi i za ovu, s izuzetkom načina merenja koje se ovde izvodi u printerskim tačkama čija je veličina **1/72 inča**. Većina fontova se stvara tako što se meri tačkama, pa je ovo sasvim prirodan način odabiranja visine za sve koji su kada radili sa fontovima. Tačka je kao merna jedinica za visinu karaktera uvedena u grafičkoj industriji mnogo pre pojave računara i štampači su se jednostavno uklopili u postojeći sistem. To znači da pojam tačke kao merne jedinice ne treba shvatiti kao veličinu iglice štampača, već kao konvencijski naziv za dužinu jednog sedamdeset drugog dela inča.

ZADAVANJE NAGIBA TEKSTA
Code: 0D Set Character Baseline Vector

```
WORD vst_rotation(handle angle)
     WORD handle
     WORD angle
```

Nagib teksta se zadaje tako što se odredi ugao osnovne linije (parametrom **angle**). Ovako zadat ugao važi za sva ispisivanja teksta pomoću VDI funkcije izdavanja.

Ugao se meri u smeru suprotnom kazaljki na satu od horizontalne linije i to u desetim delovima stepena. Tako ugao 0 znači standardni horizontalni ispis a ugao 900 znači ispis po vertikali. Ovakvi ispisi se često koriste pri izradi tehničkih crteža.

IZBOR FONTA
Code: 15 Select Character Font

```
WORD vst_font(handle font)
     WORD handle
     WORD font
```

Ovom funkcijom se bira tekući font kojim će se raditi sva izdavanja teksta. Font se bira navođenjem **indeksa** (neki ga nazivaju i **identifikacionim brojem**). Indeks 1 je rezervisan za sistemski font koji ne mora postojati na svim radnim stanicama, odnosno uređajima. Svi ostali fontovi su eksterni i moraju se učitati **vst_load_fonts** funkcijom. Indeks fonta se nalazi **vrl_name** funkcijom koja pripada klasi funkcija upita.

Funkcija vraća indeks izabranog fonta ili kod greške.

BOJA TEKSTA
Code: 16 Set Text Colour

```
WORD vst_color(handle color_index)
     WORD handle
     WORD color_index
```

Postavlja boju teksta. Uvek postoje barem dve boje (0 i 1). Postavljena boja ostaje važeća za sva ispisivanja teksta i to sve dok istom ovom funkcijom ne bude promenjena. Funkcija vraća vrednost postavljenog koda ili kod greške.

SPECIJALNI EFEKTI
Code: 6A Set Text Special Effects

```
WORD vst_effects(handle effect)
        WORD handle
        WORD effect
```

Parametar **effects** je bit mapiran. Svaki od njegovih šest najnižih bitova označava postojanje ili nepostojanje jednog specijalnog efekta. Ukoliko je **effects** 0×0000 nema nikakvih efekata i kaže se da se tekst ispisuje u **normal** modu. Zadavanje željene kombinacije efekata najlakše se izvodi ako se definiše šest heksadekadnih konstanti (za one koji su zaboravili: heksadekadna konstanta se u C-u piše kao 0×**HHHH**, gde su **HHHH** četiri heksadekadne cifre) koje se potom kombinuju logičkim operatorom OR (u C u se to piše jednostavno |).

BOLD	-0×0001 /* zadebljani	*/
LIGHT	-0×0002 /* zbledenli	*/
ITALIC	-0×0004 /* iskošeni */	
UNDERLINED	-0×0008 /a podvučeni	*/
OUTLINED	-0×0010 /* konturni	*/
SHADOWED	0×0020 /* sa senkom	*/

Funkcija vraća bit-mapiran broj u koma su postavljeni svi bitovi za odabranu kombinaciju efekata. Pošto nisu na svim uređajima mogući svi efekti, dobro je testirati ovu vrednost vrednost kako bi se utvrdilo da li su svi efekti prihvaćeni.

PODEŠAVANJE TEKSTA
Code: 27 Set Graphics Text Alignment

```
VOID vst_alignment(handle hor_in vert_in
        hor_out vert_out)
        WORD handle
        WORD hor_in vert_in
        WORD hor_out WORD *vert_out
```

U funkcijama izdavanja teksta (**v_gtext**, **v_justified**) navode se koordinate početka teksta. Ovom funkcijom se precizno određuje na koj deo teksta se odnose **x** i **y** koordinata. Za **x** postoje tri mogućnosti, a za **y** čak šest. Parametri **hor_in** i **vert_in** zadaju kodove koji odgovaraju izabranom podešavanju, dok se preko pointera **hor_out** i **vert_out** vraćaju podaci o tome kako je podešavanje izvršeno. Ako se pogrešno zadaju parametri, izvršiće se postavljanje inicijalnih vrednosti koja su u sledećoj labeli ne značene zvezdicom (*)

hor_in

0*	leva ivica
1	sredina
2	desna ivica

vert_in

0*	osnovna linija
1	linija polovine
2	linija podizanja
3	vrsna linija
4	linija spuštanja
5	donja linija

Atributi popunjavanja

NAČIN POPUNJAVANJA
Code: 17 Set Fill Interior Style

```
WORD vsf_interior(handle style)
        WORD handle
        WORD style
```

Ovom funkcijom se određuje osnovni način popunjavanja kontura. Postoji ukupno pet načina. Kod izabranog načina se predaje parametrom **style**. Raspored kodova dat je sledećom labelom (koja se obično nalazi u nekom H fajlu kao deo «define deklaracija)

style	kod
S_HOLLOW	0
FIS_SOLID	1
FIS_PATTERN	2
FIS_HATCH	3
FIS_USER	4

FIS_HOLLOW — „prazno" popunjavanje tj. popunjavanje bojom pozadine

FIS_SOLID — „puno" popunjavanje tj. ispunjavanje čitave konture zadatom bojom bez ikakvog uzorka

FIS_PATTERN — popunjavanje uzorkom koji se zadaje funkcijom **vsf_style**

FIS_HATCH — popunjavanje šrafurom koja se zadaje kao i uzorak

FIS_USER — popunjavanje korisnički zadatim uzorkom. Funkcija vraća vrednost postavljenog koda. Ukoliko je kod zadat nepravilno, postavlja se inicijalni kod 0

BIRANJE UZORKA
Code: 18 Set Fill Style Index

```
WORD vsf_style(handle style_index)
        WORD handle
        WORD style_index
```

Ukoliko je funkcijom **vsf_interior** izabran način popunjavanja **FIS_PATTERN** ili **FIS_HATCH**, ova funkcija treba da bude pozvana kako bi se odabrao indeks uzorka (**Pattern**) ili šrafure (**Hatch**). Na slici 11 vidi se kako izgledaju uzorci i šrafure i koji su im indeksi. Ti indeksi se zadaju parametrom **style-index**. Ukoliko se zadati indeks ne nalazi u dozvoljenom opsegu, postavlja se indeks 1 kao inicijalni.

Slika 10

Uzorci 1 do 8 su u stvari rasteri poređani po rastućem intenzitetu. Njihovom upotrebom se postižu različiti intenziteti boje kojom se popunjava. Odabiranje uzorka 8 je isto kao i odabiranje **FIS_SOLID** načina popunjavanja. Šrafure se sastoje samo od kombinacija horizontalnih, vertikalnih i dijagonalnih pravih linija, gradeći uzorke koji se obično koriste pri izradi tehničkih crteža

BOJA ZA POPUNJAVANJE
Code: 18 Set Fill Colour Index

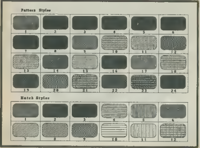
```
WORD vsf_color(handle color_index)
        WORD handle
        WORD color_index
```

Postavlja boju za popunjavanje kontura. Uvek postoje barem dve boje (0 i 1). Postavljena boja ostaje isteka za sva popunjavanja i to sve dok istom ovom funkcijom ne bude promenjena. Funkcija vraća vrednost postavljenog koda ili kod greške.

VIDLJIVOST RUBA
Code: 68 Set Fill Perimeter Visibility

```
WORD vsf_perimeter(handle per_vis)
        WORD handle
        WORD per_vis
```

Pri izdavanju popunjenih geometrijskih likova (uglavnom funkcijama GDP grupe) potrebno je odrediti da li će rub tj. kontura biti vidljiva ili ne. Ako je **per_vis**=**0** kontura **nije vidljiva**, a inače je vidljiva.



POSTAVLJANJE KORISNIČKOG UZORKA
Code: 70 Set User Defined Fill Pattern

```
VOID vsf_updat(handle pfill_pat planes)
        WORD handle
        WORD pfill_pat[ ]
        WORD planes
```

Korisnički definisan uzorak je u stvari bit mapiran blok od 16 reči, što daje uzorak veličine **16 × 16** tačaka. Ukoliko se popunjavanje vrši u načinu zamene (**MD_REPLACE** kod=1) može se koristiti više ravni uzoraka, naj više onoliko koliko ravni boja postoji na zadatoj radnoj stanici. Parametar **planes** daje broj ravni, a niz **pfill_pat[planes*16]** sadrži onoliko blokova koliko ima ravni. Da bi se sve ovo koristilo, funkcijom **vsf_interior** mora se postaviti **FIS_USER** način popunjavanja.

Rasterske funkcije

(Raster Functions)

Glavna namena rasterskih funkcije je manipulisanje blokovima memorije u kojima se bit-mapirano nalaze delo-

Programiranje laserskih štampača (3)

Rad sa fontovima

Najveći deo komunikacije sa štampačem svodi se, sasvim prirodno, na ispisivanje teksta. Zato ćemo ovoga meseca upoznati komande PCL-a koje operišu sa fontovima.

Dejan Ristanović

U ROM *LaserJet*-a ugrađena su svega tri fonta: *Courier 10 Pitch*, *Courier Bold 10 Pitch* i *Line Printer 16.66 Pitch*. U ROM-ove raznih kompatibilnih štampača drugih proizvođača ugrađuju se i drugi fontovi koje, pošto u njih nisu ugrađene YU slove, koristimo uglavnom za listinge i manje tabele. Slično bi moglo da se kaže i za fontove u kartridžima – izbor je prilično veliki (svaki proizvođač se dodušé trudi da se u njegov štampač ubacuju isključivo njegovi moduli), ali naših slova nigde nema. Zato ćemo se u okviru ove serije napisa uglavnom baviti takozvanim *soft* fontovima: datotekama u koje su upisani oblici slova i koje se pre štampanja prosleđuju laserskom štampaču i upisuju u njegov RAM. Zgodna osobina ovih fontova je što ih pre slanja štampaču možemo proizvoljno prepravljati (npr. pomoću *Fontgen*-a ili *Soft Craft Font Editor*-a) i tako zameniti neke manje potrebne slove našim ćčđšž. Korišćenje *soft* fontova osim toga ne zahteva nabavku posebnog hardvera (ROM odnosno EPROM moduli sa fontovima nisu ni malo jeftini). Tržište je vrlo široko (*Ventura*, *Pagemaker*, *WordPerfect*...) a mogućnost kombinovanja potpuna. Loša strana ovih fontova je što zauzimaju RAM štampača (originalni *LaserJet* II sa 512 K RAM-a, na primer, uopšte ne može da primi font veličine 72 pt!) i što njihovo učitavanje zahteva određeno vreme.

Učitavanje fontova zahteva poznavanje određenog skupa komandi i zadavanje brojnih parametara kojima ćemo se baviti sledećeg meseca. Samo učitavanje, naravno, nije dovoljno – treba upoznati komande kojima se neki od fontova aktiviraju.

Najjednostavnije bi bilo kada bi svaki font imao broj i kada bi postojala komanda „od sada štampaj fontom broj NN". Ovakvo komandovanje, međutim, *Hewlett Packard* iz nepoznatih razloga nije usvojio – kod prvog *LaserJet*-a fontovima uopšte nisu mogli da se dodeljuju identifikacioni brojevi, dok je na *LaserJet*+ i *LaserJet* II dodeljivanje brojeva moguće ali i dalje ne predstavlja primarni metod upravljanja ispisom.

Kako se onda aktiviraju fontovi? Slanjem opisa: kažemo štampaču nešto poput „treba mi helvetika 10 pt, PC set karaktera, malo crnja od normalnog". Štampač pronađe ono što najviše liči na naš opis i štampa u tom fontu. Kada bi u ROM-u i RAM-u *LaserJet*-a bilo nekoliko stotina fontova, ovakvoj metodologiji se ne bi moglo ništa zameriti; kada je, međutim, broj fontova ograničen na 32 (na jednoj strani se može pojaviti svega 16 fontova – na *LaserJet*+ su ova ograničenja još oštrija), RAM na 0.5 ili 1.5 megabajta a broj fontova u ROM-u na tri, čitava koncepcija pada u vodu – korisnik učitava samo fontove koji su mu zaista neophodni i uvek tačno zna koji će od njih aktivirati. Zadavanje fonta preko karakteristika, međutim, moramo da upoznamo jer se radi o metodu koji ćemo vrlo često primenjivati.

Parametri fontova

Osnovne karakteristike prema kojima se bira font su:

Orijentacija: *LaserJet* oštro razlikuje *portrait* i *landscape* orijentaciju ispisa, pa se za svaku od njih moraju koristiti i odgovarajući fontovi – *portrait* fontovima ne može se pisati na *landscape* stranici i obratno. Zanimljivo je dodati da neki *LaserJet* kompatibilni printeri (npr. *Mannesmann Tally 910*) automatski rotiraju fontove iz jedne orijentacije u drugu, dok se pri radu sa većinom PCL štampača ovaj posao mora poveriti kompjuteru i programu kao što je PORTOLAN iz paketa *Ventura Publisher*.

Komande za izbor *Landscape* fonta svodi se na komandu za izbor *landscape* stranice – Esc &l0O (malo l, cifra 0, veliko slovo O) bira *portrait* a Esc &l1O (malo l, cifra 1, veliko slovo O) *landscape* font.

Set simbola: određuje koji ASCII kod proizvodi koji oblik. Što se kodova 32–126 tiče stvar je prilično standardizovana – zna se da je 33 blanko, 48 nula, 65 veliko A, 97 malo a i tome slično. Zaljubljenici u (već prilično zastareli) sedmobitni YUSCII unekoliko menjaju ovaj raspored (kod 91 je, na primer, njihov znak [nego slovo Š), ali se ta promena obično ne regi

slika 2

```
10 E$=CHR$(27)
20 LPRINT E$ "&l12C                    line spacing 12/48   (8 tačaka)
30 LPRINT E$ "(s8V"                    visina fonta 8 tačaka
40 LPRINT "Visina  8 tačaka
50 LPRINT E$ "(s10V"                   visina fonta 10 tačaka
60 LPRINT "Visina 10 tačaka
70 LPRINT E$;"(s12V"                   visina fonta 12 tačaka
80 LPRINT "Visina 12 tačaka
90 LPRINT CHR$(12                      Form Feed
100 END
```

OZNAKE FAMILIJA SLOVA

Familija slova	Kod	Familija slova	Kod
Line Printer	0	Zapf Humanist (Optima)	17
Pica	1	Garamond	18
Elite	2	Cooper	19
Courier	3	Coronet	20
Helvetica (Swiss)	4	Broadway	21
Times Roman (Dutch)	5	Bodoni	22
Gothic	6	Century Schoolbook	23
Script	7	University Roman	24
Prestige	8	Avant Garde Gothic	25
Caslon	9	Korinna	27
Orator	10	Bitstream Charter	28
Presentation	11	Cloister Black	29
Helvetica Condensed	12	Galliard	30
Futura	14	Futura Book	136
Palatino (Zapf Call.)	15	Futura Light	146
Souvenir	16	Helvetica Light	148

slika 3

OZNAKE SETOVA ZNAKOVA

Set simbola	Kod	Set simbola	Kod
Sedmobitni		**Osmobitni**	
ASCII JIS	0K	Roman-8	8U
ASCII US	0U	Kana 8	8K
Chinese ISO	2K	Math-8	8M
Norwegian ISO v1	0D	ANSI/Windows	9U
Norwegian ISO v2	1D	PC-8	10U
English	1E	Denmark PC 7	11U
English 2	2E		
French ISO	0F		
French ISO 2	1F		
German HP	0G	**Ne-alfabetski**	
German ISO	1G		
IRV	2U	Line Draw	0B
Italian	0I	Math Symbols	0A
Japanese	1J	Matha	0O
Latin ISO	0N	Mathb	1O
Portuguese ISO/IBM	5S	Math7	0M
Portuguese ISO	4S	PiFont	15Q
Roman Extension	0E	PiFonta	2Q
Spanish HP	1S	OCR A	0O
Spanish ISO/IBM	6S	OCR B	1O
ISO	2S	Bar 3 of 9	0Y
Swedish	3S	EAN/UPC	8Y
Swedish/Finnish	0S		
US Legal	1U		

slika 1

struja u opisu fonta. Među kodovima 128—255 (kod PCL-a se obično govori o kodovima 160—255 pošto se u rasponu 128—159 na *LaserJet+* ne mogu naći znakovi, kod *LaserJet-a II* ovo ograničenje ne postoji) vlada pravi haos: PC set, Ventura set, Symbol set, Math set... *Hewlett Packard* je propisao oznake setova znakova sa slike 1, pri čemu se najčešće koriste PC 8 (IBM PC set znakova) i *Roman 8*. Kada budemo opisivali zaglavlja fonta, videćemo da bilo koja kombinacija cifra-slovo može da predstavlja set znakova: nema, dakle, nikakve prepreke da formiramo fontova zvane 8J (8Y je zauzelo) koji bi predstavljali *Roman-8* set znakova sa YU slovima. Ovako nešto međutim malo ko radi.

Set znakova se određuje komandom Esc (<oznaka>: primetimo da format ove komande u suštini odudara od standardnog oblika PCL sekvenci *(ESC grupa argument znak)*, ali je zahtevom da se ime seta znakova sastoji od brojke i slova formalna kompatibilnost sačuvana.

Proporcionalnost određuje da li je font proporcionalan ili nije — kod neproporcionalnog fonta sva su slova jednako široka, dok je kod proporcionalnog fonta W znatno šire nego I. Proporcionalni font se zahteva sa Esc (s1P a neproporcionalni sa Esc) s0P.

Sva tri fonta ugrađena u ROM *LaserJet-a* su neproporcionalna, ali je zato najveći deo *soft* fontova proporcionalan.

Širina slova (pitch): kod proporcionalnih fontova *pitch* označava širinu svakog znaka izraženu u broju znakova po inču. Tipična vrednost su 10, 12 i 16.66 CPI. Osnovni font iz ROM-a *LaserJet-a* i većina kompatibilnih laserskih štampača je neproporcionalan, dok je širina slova 10 CPI.

Kod proporcionalnih fontova svako slovo je različito široko, što znači da sam smisao pojma *pitch* promenjen: usvojeno je da umesto širine slova *pitch* predstavlja širinu blanko simbola (ASCII kod 32) u fontu.

Širinu slova zahtevanog fonta definišemo sekvencom Esc (s#H, gde smo sa # označili širinu: broj može da bude ceo ili decimalan, na primer Esc (s16.66H. Komandom Esc &k#S takođe utičemo na širinu slova, ali se radi o „nasilnoj" promeni širine blanko simbola u cilju uravnavanja desne ivice teksta. Ovu komandu ćemo upoznati nešto docnije.

Visina slova se, kao i širina, izražava u decimalnim tačkama i predstavlja parametar fonta, za razliku od prošlog meseca pomenutog *line spacing-a* (komanda Esc &l#C) koji označava razmak između redova i **obuhvata** i visinu slova. Program sa slike 2 definiše fiksan razmak između redova od 12/48 inča (nova jedinica, pozdravite *Hewlett Packard*) odnosno 18 decimalnih tačaka (decimalna tačka je, sećamo se, 1/72 inča), a zatim bira fontova visine 8, 10 i 12 tačaka; u zavisnosti od kertridža i učitanih *soft* fontova, rezultat izvršenja programa može da bude različit.

Visina slova se, kao što vidimo, bira sekvencom Esc &l#C gde je „tarabom" označen argument izražen u decimalnim tačkama.

Kurziv (italics): Aktiviranje kurzivnih *(italic)* slova postižemo sekvencom Esc (s1S, dok se standardnom (vertikalnom, *upright*) ispisu vraćamo sa Esc (s0S.

Zacrnjenje (Boldface, Stroke Weight) obezbeđuje ispisivanje važnijeg teksta polucrnim **(bold)** slovima. Autori PCL-a su odredili da zacrnjenje bude u intervalu [−7, 7], pri čemu 0 označava normalna slova, pozitivni brojevi zatamnjenje, a negativni svetlije *(light)* fontove. Praksa nije sledila teoriju — na tržištu ćete naći fontova čije je zacrnjenje 0, +3 (polucrno) i... praktično ništa više.

```
<Esc> (## ##       1B 28 ## ##        Potreban set znakova sa oznakom ## ## (brojka
                                      i slovo) kao primarni.
<Esc> )## ##       1B 29 ## ##        Potreban set znakova sa oznakom ## ## (brojka
                                      i slovo) kao sekundarni.
<Esc> (s1P         1B 28 73 31 50     Potreban proporcionalan font kao primarni.
<Esc> (s0P         1B 28 73 30 50     Potreban neproporcionalan font kao primarni.
<Esc> )s1P         1B 29 73 31 50     Potreban proporcionalan font kao sekundarni.
<Esc> )s0P         1B 29 73 30 50     Potreban neproporcionalan font kao sekundarni.
<Esc> (s##H        1B 28 73 ## 48     Potreban primarni font čiji su znaci široki ##
                                      decimalnih tačaka.
<Esc> )s##H        1B 29 73 ## 48     Potreban sekundarni font čiji su znaci široki
                                      ## decimalnih tačaka.
<Esc> (s##V        1B 28 73 ## 56     Potreban primarni font čiji su znaci visoki ##
                                      decimalnih tačaka.
<Esc> )s##V        1B 29 73 ## 56     Potreban sekundarni font čiji su znaci visoki
                                      ## decimalnih tačaka.
<Esc> (s1S         1B 28 73 31 53     Potreban kurzivni font kao primarni.
<Esc> (s0S         1B 28 73 30 53     Potreban ne-kurzivni font kao primarni.
<Esc> )s1S         1B 29 73 31 53     Potreban kurzivni font kao sekundarni.
<Esc> )s0S         1B 29 73 30 53     Potreban ne-kurzivni font kao sekundarni
<Esc> (s##B        1B 28 73 ## 42     Potreban primarni font čija je zacrnjenja ##.
                                      Argument je između -7 i +7.
<Esc> )s##B        1B 29 73 ## 42     Potreban sekundarni font čije je zacrnjenja ##.
                                      Argument je između -7 i +7.
<Esc> (s##T        1B 28 73 ## 54     Potreban primarni font iz familije ##.
<Esc> )s##T        1B 29 73 ## 54     Potreban sekundarni font iz familije ##.
<Esc> *c##D        1B 2A 63 ## 44     Radiće se sa identifikacionim brojem ##. Broj
                                      je između 0 i 32767.
<Esc> *c0F         1B 2A 63 30 46     Briše sve soft fontove i njima dodeljene id.
                                      brojeve kao i id. brojeve svih fontova iz
                                      ROM-a. Pri brisanju se ne obraća pažnja na to
                                      da li je font (id. broj) označen kao stalan ili
                                      kao privremen.
<Esc> *c1F         1B 2A 63 31 34     Briše sve privremene fontove kao i privremene
                                      id. brojeve dodeljene ROM fontovima.
<Esc> *c2F         1B 2A 63 32 34     Briše trenutno izabrani font.
<Esc> *c3F         1B 2A 63 33 34     Briše poslednji karakter koji je definisan u
                                      fontu koji je trenutno aktivan.
<Esc> *c4F         1B 2A 63 34 34     Proglašava izabrani font privremenim.
<Esc> *c5F         1B 2A 63 35 34     Proglašava izabrani font permanentnim.
<Esc> *c6F         1B 2A 63 36 34     Dodeljuje tekući id. broj tekućem fontu.
<Esc> (##X         1B 28 ## 58        Aktivira font čiji je id. broj ## i proglašava
                                      ga za primarni.
<Esc> )##X         1B 29 ## 58        Aktivira font čiji je id. broj ## i proglašava
                                      ga za sekundarni.
<SI>               0F                 Aktivira primarni set znakova.
<SO>               0E                 Aktivira sekundarni set znakova.
```

Izbor fonta odgovarajućeg zacrnjenja * postižemo sekvencom Esc (s*B

Familije slova (typeface) je parametar na osnovu koga su laserski štampači zaradili dobar deo svoje reputacije — uz sasvim umerene ulaganje u *soft* fontove možete da štampate helvetikom tajmsom garmondom Komanda Esc (s*T zahteva da dalji ispis bude slovima tipa * pri čemu su kodovi (0—255) formiraju u skladu sa slikom 3 Radi se o familijama slova koji je *Hewlett Packard* ozvaničio ali mnogi drugi proizvođači fontova rezervišu dodatne kodove za svoja dizajnerska remek-dela što ponekad stvara haos Kod nas je uobičajeno da YU Helvetica i YU Times nose brojeve standardne Helvetice i Times-a (4 odnosno 5) dok se za ćirilicu u Helvetici i Tajmsu rezervišu kodovi 252 i 254

Prioritetni kriterijumi

Ukoliko po prijemu duge kontrolne sekvence *LaserJet* pronađe font koji potpuno zadovoljava zahteve stvar je sasvim jednostavna — font će biti izabran Ukoliko međutim ni jedan od raspoloživih fontova ne zadovolji sve zahteve *LaserJet* prema onome što piše u dokumentaciji *aktivira font koji je najsličniji traženom* U nekim slučajevima vredi znati kako ovo odlučivanje teče što je najlakše razmotriti na primeru

Pretpostavimo da su u RAM *LaserJet*-a učitani sledeći fontovi

Parametar	Font 1	Font 2	Font 3	Font 4
Orijentacija	portrait	portrait	portrait	portrait
Set simbola	Roman-8	USASCII	USASCII	USASCII
Proporc.	prop	prop	neprop	prop
Širina slova	10	10	17	10
Visina slova	12	12	8	12
Kurziv	ne	ne	ne	da
Zacrnjenje	0	0	0	0
Familija slova	Times	Times	Helvetica	Helvetica

Štampaču je poslata komanda **Esc &l0O Esc (0U Esc (s1P Esc (s10H Esc &l12C Esc (s0B Esc (s0S Esc (s4T** (zahvaljujući metodama kombinovanja istorodnih sekvenci o kojima smo govorili pre mesec dana ova komanda bi mogla da se skrati ali ovako je jasnije) čime je zahtevan font sledećih karakteristika

Parametar	Vrednost
Orijentacija	portrait
Set simbola	USASCII
Proporc	prop
Širina slova	10
Visina slova	12
Kurziv	ne
Zacrnjenje	0
Familija slova	Helvetica

Najpre se ispituje orijentacija — tražen je *portrait* font svi postojeći fontovi su *portrait* pa je test položen Font 1 će prvi otpasti pošto se zahtevani set simbola ne poklapa sa realnim Sledeći se izuzima font 3 pošto je neproporcionalan što znači da su „u igri" ostali fontovi 2 i 4 Oba su odgovarajuće širine i visine ali je jedan kurzivni a drugi iz pogrešne familije Tvorci PCL-a su kurzivu dodelili viši prioritet pa će font 4 biti eliminisan a font 2 izabran — štampaće se tajms umesto helvetike ali su dati ostali parametri korektni

Pokušajmo sada sa malo težim primerom sekvencom **Esc &l0O Esc (0U Esc (s1P Esc (s14H Esc &l10C Esc (s0B Esc (s1S Esc (s6T** je zahtevan font sledećih karakteristika

Parametar	Vrednost
Orijentacija	portrait
Set simbola	USASCII
Proporc	prop
Širina slova	14
Visina slova	10
Kurziv	da
Zacrnjenje	0
Familija slova	Gothic

„Rezonovanje" *LaserJet*-a teče otprilike ovako font 1 nema odgovarajući set simbola font 3 je neproporcionalan pa opet ostaju fontovi 2 i 4 Ni jedan od njih ne zadovoljava traženu širinu ili visinu slova pa se po ovom kriterijumu ne može izvršiti izbor ispituje se stil ispisa i font broj 2 (nije kurzivan) otpada Biće izabran font 4 bez obzira na pogrešnu familiju slova

Ne treba imati mnogo programerskog iskustva da bi se zaključilo kako „gađanje" o fontovima teško može da proizvede dobre rezultate — programer uvek želi da bude siguran da će neki konkretan font biti izabran Zato je počev ši od *LaserJet+* predviđeno dodeljivanje identifikacionih brojeva fontovima

Identifikacioni broj je bilo koji ceo broj između 0 i 32767 Ukoliko se radi o *soft* fontu broj mu dodeljujemo pre nego što smo ga uopšte definisali ili poslali štampaču — ovu funkciju automatski obavljaju programi LOADFONT i AUTODL iz paketa *Ventura Publisher* Ukoliko se radi o nekom od fontova iz ROM-a ili kertridža najpre ćemo ga izabrati korišćenjem sekvence slične nekoj od pomenutih (dajzik program sa slike 4 ilustruje izbor osnovnih fontova iz ROM-a *LaserJet*-a) a onda izvršiti Esc *c*D gde je * broj koji dodeljujemo fontu — na ovaj način je identifikacioni broj * proglašen aktuelnim Zatim izvršavamo Esc *c6F i tako aktuelnom fontu dodeljujemo aktuelni broj dovršavajući kreiranje takozvanog privremenog *(temporary)* identifikacionog broja — prva sledeća *reset* komanda će anulirati dodeljivanje Ukoliko želimo da identifikacioni broj „preživi" svaki *reset* Esc * c6F ćemo zameniti sa Esc * c6F5F Svi identifikacioni brojevi i *soft* fontovi se jasno gube kada isključimo laserski štampač

Komanda Esc *c*F obezbeđuje različite operacije sa fontovima u zavisnosti od vrednosti argumenta *

Esc *c0F briše sve *soft* fontove i njima dodeljene identifikacione brojeve kao i identifikacione brojeve svih fontova iz ROM-a. Pri brisanju se ne obraća pažnja na to da li je font (i identifikacioni broj) označen kao stalan ili kao privremen

Esc *c1F briše sve privremene *(temporary)* fontove kao i privremene identifikacione brojeve dodeljene ROM fontovima

Esc *c2F briše trenutno izabrani font.

Esc *c3F briše poslednji karakter koji je definisan u fontu koji je trenutno izabran Ova funkcija se koristi izuzetno retko — obično se pokazuje da je nepotreban kompletan font a ne samo neka slova u njemu

Esc *c4F proglašava izabrani font privremenim

Esc *c5F proglašava izabrani font permanentnim

Esc *c6F dodeljuje tekući identifikacioni broj tekućem fontu Dodeljivanje je privremeno da bi dodeljen broj preživeo *reset* treba izvršiti Esc *6f5F

Brisanje fonta je u izvesnom smislu konfliktna operacija — čin brisanja označava da nam je memorija koju je zauzimao font potrebna za nešto drugo i to odmah Ukoliko se u baferu štampača nalazi neki tekst koji sadrži font koji se briše taj tekst mora odmah da se štampa jer će po brisanju fonta biti kasno Zato će *LaserJet* ukoliko se na tekućoj strani nalazi nešto teksta ispisanog obrisanim fontom najpre štampati stranu (ako ona nije kompletna sami smo krivi) i tek onda početi da prima komande računara Ukoliko na tekućoj strani nema nikakvog teksta pisanog fontom koji se briša *LaserJet* će proveriti da li se u baferu nalaze još neke strane čije je štampanje u toku ako ih ima prijem komandi od računara će biti zaustavljen dok se ove stranice ne ispišu Ova diskusija objašnjava izvesnu anomaliju koju je sigurno primetio svaki korisnik *Venture* štampa se nekoliko strana a onda lampica miruje dok se kompletan bafer ne isprazni To je naime trenutak u kome je interni menadžer fontova zaključio da neke od fontova treba ukloniti iz RAM a štampača

Pošto je fontu dodeljen identifikacioni broj, ispisivanje odgovarajućim slovima iniciramo sa Esc (*X gde je * identifikacioni broj potrebnog fonta

Ostalo je još da pomenemo prvi (!?) metod za izbor fontova koji se izuzetno retko koristi Konstruktori PCL a su obezbedili definisanje dva fonta od kojih će se jedan zvati „primarni" a drugi „sekundarni" Pošto su fontovi definisani sa SI aktiviramo prvi a sa SO drugi — zgodno je što fontove zamenjujemo slanjem jednog jedinog bajta ali je nezgodno što nikada nismo sigurni koji je font trenutno aktivan (komunikacija sa laserom je jednosmerna)

Sledećeg meseca upoznaćemo komande koje se koriste za slanje fontova štampaču *(download)* i strukturu SFP datoteka koje se isporučuju uz program *Ventura Publisher*

slika 4

```
 10 REM
 20 REM Fontovi ngradeni n  LaserJet II
 30 REM         |PC karakter set|
 40 REM
 50 ESC$=CHR$(27)
 60 REM Conrier 10 pitch normal
 70 LPRINT ESC$;"&l0O";ESC$;"|10U"
 80 LPRINT ESC$;"(s0p10h12v0s0b3T";
 90 LPRINT "Courier 10 pitch normal"
100 REM Conrier 10 pitch bold
110 LPRINT ESC$;"&l0O",ESC$,"|10U";
120 LPRINT ESC$ "(s0p10h12v0s3b3T"
130 LPRINT "Conrier 10 pitch bold"
140 REM Line Printer normal
150 LPRINT ESC$,"&l0O",ESC$,"|10U",
160 LPRINT ESC$;"(s0p16.6h8 5v0s0b0T"
170 LPRINT "Line printer normal"
180 LPRINT CHR$|12|;
190 END
```

Disk jedinice softver

Pređimo na logiku

U prošlom broju smo se bavili fizičkim karakteristikama mašinske opreme diska — brojem staza ili cilindara po ploči, brojem ploča, veličinom sektora u bajtovima, brojem sektora, faktorom redosleda itd. Te karakteristike su nepromenljivi elementi hardvera. Međutim, postoji još čitav niz drugih karakteristika diska vezanih za operativni sistem i način na koji on organizuje i nalazi podatke na disku. Tekst je rađen prema dokumentaciji koja se isporučuje uz softverski paket Norton Utilities 5.0.

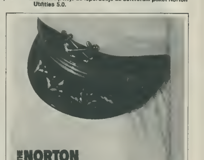
Zbog brzine i efikasnosti i korišćenja određenih bajtova u ogromnom moru megabajtova od 20 40 pa i mnogo više, jasno je da je neophodno da operativni sistem napravi neka direktorijume i indekse koji će mu reći gde je šta, šta se na kojim delovima diska nalazi koji su delovi slobodni pa čak i koji se delovi ne smeju koristiti zbog fizičkog oštećenja. Način na koji su te informacije organizovane na disku naziva se logički format diska (logical format) a proces upisivanja raznih direktorijuma i indeksa koji podržavaju tu organizaciju naziva se logičko formatizovanje. Kao što smo ranije pomenuli kada formatizujete tvrdi disk koristeći DOS FORMAT naredbu (ili Safe Format program) vi, u stvari izvodite logičko formatizovanje jer je fizičko formatizovanje već obavljeno. Međutim kod disketa DOS FORMAT naredba prvo izvodi fizičko formatizovanje čemu automatski sledi logičko formatizovanje. S druge strane Safe Format će izvesti samo fizičko formatizovanje ako je potrebno. Bez obzira koju vrstu diska koristimo — tvrdi ili meki fiksni ili zamenljiv od 360K ili 60 megabajta — DOS uvek pribegava istom logičkom formatizovanju koje organizuje disk u četiri glavna područja. **početni zapis** (boot record) **tabela za smeštanje datoteka** (FAT — file allocation table) **glavni direktorijum** (root directory) i **područje podataka** (data area) (Na tvrdom disku koji se može podeliti upotrebom raznih operativnih sistema postoji još i peto područje nazvano **tabela particija** — Partition Table o kojoj ćemo kasnije govoriti)

Početni zapis

DOS ov početni zapis uvek zauzima prvi sektor prve staze na prvoj strani diska — sektor 1 staza 0 strana 0 (u stvari to je isključivo slučaj kod disketa budući da je kod tvrdih diskova prvi sektor rezervisan za tabelu particija) Početni zapis je upravo ono što mu ime govori — sadrži početni kod (boot code) koji se prenosi u računar i odatle aktivira ostali deo operativnog sistema

Pored tog malog početnog programa, početni zapis sadrži još nešto što je od vitalnog značaja za operativni sistem listu ključnih karakteristika diska. Ova lista između ostalog obuhvata broj bajtova po sektoru ukupan broj sektora na disku broj sektora po stazi i broj glava. Zbog važnosti ove tabele početni zapis se upisuje u sve diskove prilikom logičkog formatizovanja pa čak i u diskete koje ne sadrže itosistemske datoteke potrebne za automatsko butiranje diska odnosno sistema

Drugo i treće područje koja DOS organizuje prilikom logičkog formatizovanja — FAT tabela i glavni (root) katalog — koriste se zajedno da bi se imalo uvida u to gde je smeštena svaka datoteka na disku koji su sektori u tekućoj upotrebi za smeštanje datoteka i koji su sektori raspoloživi



Grupe podataka

Kada je disk već formatizovan najmanja količina podataka koja disk kontroler fizički može da očita ili upiše jeste ona koju prima jedan sektor i koja zasad kod svih DOS diskova iznosi 512 bajtova. DOS treba (što i jeste slučaj kod disketa od 1 2 megabajta se velikom gustinom) da ima uvid u stanje svakog pojedinačnog sektora — da li je u upotrebi da li je slobodan ili se ne sme koristiti zbog oštećenja. Ali pošto je količina onoga što mora biti na uvidu po svakom sektoru površine diska dosta velika na diskovima sa velikim kapacitetom DOS pribegava višesektorskim jedinicama nazvanim **grupe podataka — klasteri — (clusters)**

Pre nego što se pobliže upoznamo sa klasterima i sa tim kako DOS ima uvid u datoteke na disku bilo bi dobro da pogledamo na koje probleme DOS nailazi kada od njega tražimo da pravi, šta, sažima ili briše naše datoteke

Rad sa datotekama

Osnovni problem kod baratanja računarskim datotekama je to što se one stalno menjaju (jednu od privlačnih strana računarskih datoteka čini naravno upravo ta sposobnost da se lako menjaju) Pretpostavimo na primer da imamo datoteku dužine od 2250 bajtova i da je upisujemo u novu neupotrebljavanu disketu od 1 2 megabajta. Najmanji broj bajtova koji se može upisati ili očitati sa diska je onaj sadržan na jednom sektoru a to je upravo minimalan broj bajtova koje će DOS očitati ili upisati na disketu od 1 2 megabajta. Za našu datoteku od 2250 bajtova trebaće nam pet sektora. Prva četiri sektora će primiti 2048 bajtova što znači da će nam ostati samo 202 bajta za peti sektor. Pošto ne možemo da delimo sektor DOS će rezervisati ceo peti sektor za našu datoteku tako da će preostali deo tog sektora ostati neiskorišćen. Naša nova datoteka zauzimaće prvih pet sektora diska

Hajde sada da upišemo neku drugu datoteku dužine recimo od sedam sektora. Ona će zauzeti narednih sedam sektora u području podataka. Sad počinje zabava. Pretpostavimo da želimo da našoj prvoj datoteci dodamo 460 bajtova. Kao što se sećate od poslednjeg sektora od 512 bajtova iskorišćeno je samo 202 što znači da imamo mesta za još 310 bajtova

Ali ostalo je 150 bajtova (460—310 bajtova koje smo dodeli na kraju poslednjeg sektora). Gde ćemo ih staviti?

Ne možemo ih jednostavno staviti u prvi sektor posle naše datoteke od pet sektora jer je taj sektor već popunjen narednom datotekom (onom od sedam sektora). Možemo da pomerimo tu narednu datoteku za sektor dalje, ali to bi nam oduzelo vremena. Osim toga pomislite koliko bi to neefikasno bilo kada bi naredni **jedan megabajt** sektora bio već zauzet datotekama. Tada bismo morali da pomerimo čitav megabajt podataka samo zato da bismo upisali tričavih 150 bajtova. To nije dobro rešenje.

Tabela za smeštanje datoteke

Mnogo bolje rešenje a to je ono koje koristi DOS jeste napraviti tabelu o stanju svih sektora. Ta važna tabela nazvana FAT reći će nam da li je dati sektor slobodan ili je zauzet datotekom. Prema tome, kada treba da proširimo datoteku samo ćemo pogledati FAT, naći sledeći slobodan sektor i rezervisati ga za našu datoteku. U našem primeru prvih 5+7 sektora su zauzeti tako da je prvi slobodni sektor smešten 13 sektora od početka područja podataka. Naša datoteka, dakle, zauzima prvih pet sektora i 13-ti sektor područja podataka. Tako je naša datoteka sada **raspačana**. Početni deo je smešten u susednim sektorima na početku diska, dok je završni deo negde drugde. Međutim, to nije problem pod uslovom da možemo da nađemo način da povežemo taj završni deo datoteke sa početnim delom tako da DOS zna gde sve to da nađe.

Malopre smo jednostavnosti radi rekli da DOS drži tabelu FAT koja beleži stanje svih sektora na disku. Ali to nije sasvim tačno. Kod diska od 32 megabajta ova tabela koja ima uvid u svaki sektor obuhvatala bi 64K podataka. (32 megabajta podeljena sa 512 bajtova po sektoru.) Pošto je zbog brzog pristupa FAT smešten u memoriji dok se disk koristi, veliki diskovi bi vrlo brzo potrošili celu memoriju. Osim toga bilo bi potrebno dosta vremena da se pretraži FAT od 128K (64K podataka puta 2 bajta po podatku). Zbog toga DOS grupiše izvestan broj susednih sektora zajedno i barata sa njima kao sa jedinicom nazvanom grupa podataka (cluster). U stvari, DOS na FAT tabeli ima uvida u klastere, a ne u sektore.

Pošto klastere pravi sam DOS, on ih može odrediti tako da se sastoje iz onoliko sektora koliko on želi. Međutim, umesto da odredi ustaljenu veličinu jednog klastera, DOS ga pravi godave različitim medijumima. (Veličina klastera na datom disku određuje se prilikom logičkog formatizovanja tako da se kasnije ne menja, izuzev ukoliko se disk ponovo ne formatizuje.) AT tvrdi disk od 30 megabajta ima veličinu klastera od 4 sektora ili 2048 bajtova, dok je disketa od 1,2 megabajta formatizovana sa veličinom klastera od 1 sektora ili 512 bajtova. Ta mala veličina klastera je odraz želje da se optimalno koristi raspoloživog prostora pretpostavi brzini. Potrebno je više vremena kada se radi sa diskovima sa manjim veličinama klastera, budući da FAT tabela mora da barata sa više podataka, budući da to zahteve više operacija očitavanja i upisivanja itd. S druge strane što je veličina klastera manja to manje prostora ostaje neiskorišćeno na kraju svake datoteke.

Malopre smo razmotrili primer datoteke koja je bila duga 2250 bajtova i zauzimala 5 sektora na disku velikog kapaciteta. Pošto svaki klaster na disku velikog kapaciteta zauzima jedan sektor, ostaju nam neiskorišćeni samo bajtovi na kraju poslednjeg sektora — oko 200 bajtova. Ali šta bi bilo da smo tu istu datoteku prebacili na tvrdi disk od 30 megabajta formatizovan sa veličinom klastera od 4 sektora? DOS koristi FAT tabelu da bi imao uvid u prostor po klasterima. Pošto će disk od 32 megabajta verovatno imati veličinu klastera od 4 sektora (2048 bajtova u DOS 3.x), najmanja veličina prostora koji DOS može da zauzme na disku iznosi 2084 bajta. Naša datoteka od 2250 bajtova će zauzeti samo 2 klastera, ali pošto svaki klaster sada ima 2048 bajtova, broj neiskorišćenih bajtova na kraju poslednjeg klastera povećao se sa nekih 200 na više od 1800.

Taj neupotrebljiv prostor na kraju klastera naziva se **mrtav prostor** (slack) i on je razlog što ponekad izgleda da se datoteke smanjuju kada ih prebacujete sa tvrdog diska na diskete a povećavaju kada se prebacuju sa diskete na tvrdi disk. Prihvatamo veći mrtav prostor u većim klasterima da bismo za uzvrat dobili bolje karakteristike koje nam oni nude.

Korišćenjem programa za veličinu datoteke (File size program) u okviru paketa Norton Utilities možemo dobiti podatke o procentu mrtvog prostora svake datoteke, celog direktorijuma i celog diska. Program za veličinu datoteke može takođe da utvrdi da li ima mesta da se grupa podataka smesti na neki drugi disk uzimajući u obzir skupljanje ili širenje datoteka kada se one premeštaju sa diska sa jednom veličinom klastera na disk sa nekom drugom veličinom klastera.

Još malo o FAT-u

Vratimo se na FAT. Dosad smo rekli da DOS ima tabelu na početku svakog diska nazvanu FAT. Znamo da se FAT koristi za uvid u to koji su klasteri slobodni, koji su zauzeti i kojom datotekom. Takođe znamo da je klaster u stvari prikladno grupisanje jednog ili više sektora čiji je cilj da se smanji velika administracija potrebna za uvid u korišćenje prostora tvrdog diska. Međutim, dosad nismo proučili kako DOS pohranjuje tu informaciju na FAT.

Osnova te tabele je sušta jednostavnost. Dva bajta su (za DOS 3 i više) dodeljena svakom klasteru na disku. Redni broj svakog podatka od dva bajta direktno se poklapa sa ekvivalentno numerisanim klasterom u području podataka. Sad su nam opet potrebna objašnjenja jer još nismo naučili kako DOS numeriše klastere.

Tri načina obeležavanja

Ako se vratimo na deo gde smo proučili fizičke osobine diska, videćemo da disk kontroler komunicira sa određenim mestom koristeći sistem sa tri koordinate: broj strane (0 ili 1 za disketu), broj staze (od 0 do 39 kod dvostrane diskete sa dvostrukom gustinom) i broj sektora (od 1 do 9 kod diskete od 360K). Pošto hardver radi sa diskovima u ovom sistemu sa tri koordinate, potrebno je da na nekom nivou PC bude u stanju da radi sa diskom koristeći isti sistem — i zbog toga svi ugrađeni osnovni ulazno/izlazni sistemi (ROM-BIOS) pozivaju određena mesta na disku preko broja strane, staze i sektora. Međutim, za operativni sistem je ovaj sistem sa tri koordinate nespretan, jer se broj staza, sektora i strana menja od diska do diska. Stoga DOS smešta podatke na disk koristeći jednodimenzionalnu šemu numerisanja sektora. DOS samo numeriše sva sektore za redom počevši od sektora i strane 0, staze 0, redajući dalje sve sektore na stazi 0, zatim prelazi na sektore na strani 1, stazi 0. Tako svi sektori u jednom cilindru bivaju numerisani pre nego što DOS pređe na naredni cilindar. To smanjuje pomeranje glave, pa prema tome i vreme pristupa kada se sektori čitaju sekvencijalno (zaredom). Neprekidna šema numerisanja sektora koju DOS koristi naziva se **logičko numerisanje sektora** (logical sector numbering). Za razliku od fizičkih brojeva sektora koji počinju od 1, logički brojevi sektora počinju od 0.

Nemojte da vas zbuni činjenica da DOS numeriše sektore na disku zaredom. Ovde se ne obavlja fizičko formatizovanje i adrese se



ne upisuju na disk. Tu se u stvari vrši samo prevođenje načina na koji BIOS i disk kontroler pozivaju dati sektor na način na koji to radi DOS. Posmatrajte to kao razliku u nazivu za sektor 1 na engleskom i francuskom (sector one — premier secteur).

Kada smo završili numerisanje sektora po DOS-ovoj šemi, odmah sledi šema numerisanja klastera. Prvo preskačemo sve sektore koji su zauzeli početnim zapisom (uvek jedan sektor), FAT-om (čija veličina varira zavisno od veličine diska ali je za iste veličine diskova konstantna) i glavnim direktorijumom (koji još nismo razmatrali).

Ova tri područja, koja bivaju uvek urađena prilikom logičkog formatizovanja, smatraju se delom sistemskog područja i uključena su iz šeme numerisanja klastera.

Svi sektori izvan sistemskog područja, počevši od prvog sektora posle glavnog direktorijuma (ROOT), smatraju se delom područja podataka i uključeni su u šemu numerisanja klastera. Na disku sa veličinom klastera od 4 sektora prva četiri sektora područja podataka nazivaju se klaster 2 (numerisanje klastera počinje od 2 a ne od 0 ili 1), naredna 4 sektora su klaster 3, naredna 4 sektora su klaster 4 i tako redom od kraja diska. Disk od 30 megabajta sa veličinom klastera od 4 sektora ima oko 15230 klastera u području podataka. Poslednji broj klastera je 15231, što je za jedan više od broja klastera, budući da numerisanje klastera počinje od 2.

Vratimo se opet na FAT

Skrenimo s puta i još jednom se vratimo na FAT. Prethodno smo to učinili da bismo rekli da FAT sadrži informaciju o stanju svakog klastera na disku i da za svaki klaster na disku postoji po jedan podatak u FAT tabeli. Prva dva podatka u tabeli rezervisana su za posebnu informaciju. Treći podatak u tabeli sadrži informaciju o prvom klasteru (koji je, da ne zaboravimo, numerisan kao klaster 2). Naredni podatak sadrži informaciju o drugom klasteru (klaster 3) i tako do kraja. Poslednji podatak u tabeli govori nam o stanju poslednjeg klastera na disku. (Strogo uzevši, neke od pojedinosti koje smo sada opisali odnose se na sve diskete i samo na tvrde diskove sa jednom particijom. O particijama ćemo govoriti na kraju. Njihovo prisustvo ne menja osnovni način rada diske.) Pogledajmo koju informaciju podatak na FAT tabeli treba da nam da. Prvo, pošto i najbolji diskovi mogu imati poneki loš sektor, podatak na FAT-u za dati klaster može reći DOS-u da li negde u klasteru postoji neki loš sektor. Tako će DOS znati da nikada ne sme smestiti datoteku u taj klaster. Upravo je to jedna od informacija koje DOS upisuje kada pravi FAT prilikom logičkog formatizovanja. Kada DOS završi formatizovanje diska — upisujući

početni zapis i pravac novi FAT i glavni direktorijum — kreće da čita svaki sektor na disku. Klasteri koji sadrže sektore koji se ne mogu kako treba pročitati označavaju se u FAT-u kao loši. Ovaj proces se naziva pravljenje mape loših sektora i vrlo je važan za celovitost naših podataka.

Možda se slučajno pitate koji bi **podatak** DOS mogao da očita sa tek formatizovane i pretpostavimo prazne diskete. Dobro pitanje na koje odgovor glasi da blanko disketa nije prazna. Ona samo ne sadrži podatke koji su za nas korisni. Kada DOS ili Safe Format formatizuje disketu, on upisuje poseban bajt (F6) po celom području podataka. Kasnije u procesu formatizovanja pokušava da očita taj bajt iz svakog sektora.

Mogu postojati neki sektori koje disk kontroler nije u stanju da pročita. U tom slučaju kontroler šalje DOS-u poruku o grešci i DOS označava klaster koji sadrži taj sektor kao loš. Kontroler može da očita nekoliko drugih sektore kao da su u redu, ali oni mogu imati neispravne bajtove za proveru, što govori da je postojala greška u očitavanju podataka.

Ciklično proveravanje (Cyclic Redundancy Check). Mogli ste čuti da ga nazivaju CRC. Kada se jedan sektor ispiše, onda se na osnovu vrednosti svih bajtova upisanih u taj sektor izračunava posebna vrednost zbira za proveru (checksum) nazvana cyclic redundancy check. Ta vrednost (takođe poznata kao CRC) upisuje se na specijalno mesto na disku odmah iza sektora podataka. Kada se taj sektor kasnije očitava sa diska, očitavaju se bajtovi zbira za proveru tj. bajtovi CRC-a. To isto izračunavanje koje određuje CRC bajtove koji su bili upisani na disku odmah iza sektora podataka ponovo se izvodi ovoga puta na osnovu upravo očitanih podataka. CRC izračunat na osnovu upravo očitanih podataka upoređuje se sa ranije izračunatim i upisanim CRC na disku. Ukoliko se ova dva vrednosti ne poklapaju, to znači da očitani podaci nisu potpuno isti kao podaci koji su upisani i stoga se pojavila greška u očitavanju (read error). Tada se klaster koji sadrži taj sektor označava kao loš.

Uzgred, moglo bi vas iznenaditi sad vidite da kada kažete DOS-u da proveri kopiju diska — koristeći bilo naredbu COPY sa /V bilo naredbu SET VERIFY ON — DOS ne upoređuje izvorna podatke bajt po bajt sa odredišnim podacima; on samo očitava odredišnu datoteku i proverava nema li tamo CRC grešaka ili bilo kakve druge vrste grešaka u očitavanju. Ako CRC prođe, postoje dobri izgledi da su podaci u redu.

Dovoljno je bilo udaljavanja od teme. Završimo razmatranja FAT-a. Dosad smo videli da moramo da vodimo računa o dvema stvarima kod svakog klastera u njegovom odgovarajućem podatku na FAT-u — da li je klaster slobodan ili zauzet datotekom i da li je loš ili dobar. Ako se datoteka prelamano na povećavaju smanjuju ili na brišu, to je otprilike sve što treba da mati u FAT-u. Ali sećate li se kada smo ranije upisali kratku datoteku od pet klastera na početku diska i odmah za njom drugu nezavisnu datoteku? Kada smo zatim pokušali da dodamo neke podatke prvoj datoteci, nismo mogli da zauzmemo prvi naredni klaster zato što je već bio iskorišćen za drugu datoteku. Morali smo da posegnemo za nekim drugim klasterom negde dalje na stazi. To postavlja pitanje: kada DOS pokušava da očita prvu datoteku, kako će znati gde da nađe onaj drugi klaster koji smo napisali negde drugde na disku?

Odgovor je sasvim jednostavan. Upisali smo prvih pet klastera datoteke u prvih pet sekvencijalnih klastera na disku, zatim smo preskočili nekoliko klastera i upisali šesti klaster datoteke. Zašto ne bismo imali FAT podatak za poslednji klaster koji smo upisali — peti klaster — koji bi pokazivao koliko je klastera preskočeno do onog šestog klastera datoteke? Ako je datoteka bila upisana u klastere od 2 do 6 i ako smo zatim dodali klaster 14, možemo za šesti klaster upisati 14 u FAT podatak. Taj broj 14 za FAT podatak šestog klastera u stvari kazuje: Naći ćete naredni deo ove datoteke u klasteru 14 na disku. Da bismo bili dosledni svaki FAT podatak jedne datoteke zapisaćemo tako da pokazuje na sledeći FAT podatak datoteke tako da ćemo imati **lanac** FAT podataka gde će nas svaki podatak voditi sledećem. Pošto je sekvencijalni broj svakog FAT podatka isti kao kod klastera na koji se odnosi, da bismo locirali sve klastere datoteke samo treba da sledimo lanac FAT podatka. To je upravo procedura koju DOS sprovodi.

Ako ste me dosad pažljivo pratili, mora da ste dobro shvatili osnovu DOS-ove strukture datoteka na disku. To samo još treba da dopunimo se još nekoliko jednostavnih pojedinosti. Na primer, sada kada znamo sve klastere datoteke pomoću lanca FAT podataka, kako da odredimo početak tog lanca? U našem primeru prvi klaster naše datoteke od šest klastera bio je takođe prvi klaster područja podataka na disku — ali samo zato što sam ja tako rekao. Kako ćete sutra naći početni klaster? I da li se sećate gde sam rekao da počinje druga datoteka?

Očigledno moramo negde držati listu početnog FAT podatka za svaku datoteku. Ako znamo gde je prvi FAT podatak, znamo i gde je prvi klaster datoteke (jer su, ako se sećate, FAT podaci numerisani isto kao i klasteri na koje se odnose). Pošto svaki FAT podatak ukazuje na sledeći u lancu, možemo naći sve klastere naše datoteke.

Pored liste početnih brojeva klastera, koja nam omogućava da nađemo prvi klaster svake datoteke, potreban nam je i način da nađemo poslednji klaster svake datoteke. Tu informaciju možemo staviti na isto mesto gde je i informacija o početnom klasteru, bilo unoseći poslednji broj klastera bilo veličinu datoteke. Drugi način je da upotrebimo sam FAT smeštajući poseban bajt u FAT podatak koji odgovara poslednjem klasteru datoteke.

DOS na neki način čini i jedno i drugo. Direktorijum datoteka (o kome ćemo uskoro govoriti) beleži i dužinu svake datoteke i broj njenog početnog klastera, a poslednji FAT podatak svake datoteke je posebno označen. Naše razmatranje ovog tako značajnog FAT-a je skoro pri kraju. Znamo da je FAT tabela sa dvobajtnim podacima (kod DOS 3) gde svaki podatak odgovara svakom klasteru na disku (ranije verzije DOS-a su koristile 1.5 bajta za svaki FAT podatak. Značaj toga u praksi razmatraće se na kraju ovog odeljka). Svaki podatak nam kazuje da li je klaster na koji se odnosi raspoloživ, defektan ili već popunjen datotekom. Ako je klaster zauzet, podatak ili upućuje na naredni klaster (ili FAT podatak datoteke) ili nam kaže da je taj klaster poslednji klaster datoteke. Stoga FAT podatak može da sadrži jednu od sledeće četiri vrednosti:

Vrednost (hex)	Značenja
0000	Klaster je na raspolaganju
0002—FFEF	Klaster je zauzet datotekom (broj pokazuje na naredni klaster datoteke)
FFF0—FFF6	Rezervisan, nije iskorišćen
FFF8	Loš klaster, ne treba ga koristiti
FFF8—FFFF	To je poslednji klaster datoteke

Tako znamo da postoji informacija u direktorijumu koja pokazuje početni FAT podatak za svaku datoteku.

Glavni katalog

Direktorijum — u stvari glavni ili osnovni katalog — treći je i poslednji deo sistemskog područja svakog diska koji DOS formatizuje. Ako ste ikada pogledali katalog koji koristi NU (ili nov Directory Editor koji je sadržan u Advanced Edition of the Norton Utilities) verovatno ste otkrili da svaka stavka u katalogu sadrži mnogo više informacija nego što su broj početnog klastera i veličina svake datoteke.

Jasnije rečeno, pored veličine i broja početnog klastera svaka stavka u katalogu mora da navede ime datoteke zajedno sa brojem početnog klastera.

Posmatrano na drugi način, mogli bismo reći da postoji jedna stavka u katalogu za svaku datoteku na disku i da njen direktorijum sadrži pokazivač (pointer) za njen prvi FAT podatak u lancu tj. njen prvi klaster. Pored toga svaka stavka u katalogu od 32 bajta sadrži polja za vreme i datum, polje od jednog bajta za **atribute datoteke** i 10 bajtova koje je DOS rezervisao. Tako format svake stavke u katalogu izgleda kao u tabeli 2.

Atributi predstavljaju jednu od nekoliko posebnih odlika koje se mogu vezati za svaku datoteku kao što su: samo za čitanje (read-only), skrivena (hidden), sistemska, ime nosioca podataka (volume label), potkatalog (subdirectory) i arhiva (program Norton Utility — File Atribute — može da postavlja, menja i ponovo postavlja atribute). Atributi su kodirani u bitovima: dati atribut je postavljen ako je njegov pratni bit 1.

Atribut labela nosioca podataka je poseban slučaj. Naime, nijedna datoteka za stavku u katalogu ne postoji sa ovim atributom. Samo jedna stavka u katalogu na disku može imati atribut imena nosioca podataka i taj podatak treba da bude u glavnom katalogu diska. Labela je smeštena u ime datoteke i polja za ekstenziju podataka kataloga, koji se u ovom slučaju tretiraju kao jedno polje.

Pre nego što se upustimo u srž kataloga zastanimo malo i pogledajmo razlike između glavnog kataloga i potkataloga. Do sada smo govorili o glavnom direktorijumu kao da je on jedini katalog na disku i glavni katalog je zaista poseban posmatrano iz više uglova. Prvo, glavni katalog je jedini katalog koji se ne nalazi u području sistemskih podataka: glavni katalog je treći element sistemskog područja i uvek se locira odmah iza FAT-a. Veličina i mesto ovog kataloga su fiksirani, određeni su prilikom logičkog formatizovanja i ne mogu se kasnije menjati. Glavni katalog se i po toj fiksiranoj veličini razlikuje od potkataloga koji se mogu pravi i po potrebi povećavati, sužavati i brisati.

Veličina glavnog kataloga varira zavisno od tipa diska. Na disketi od 360K FORMAT pravi glavni katalog sa prostorom za 112 stavki, na AT tvrdom disku od 30 megabajta FORMAT pravi glavni katalog koji može da primi 512 stavki. To na primer znači da ako želite da imate više od 512 datoteka na tvrdom disku od 32 megabajta morate da napravite nekoliko potkataloga.

Svaka stavka u glavnom katalogu može se odnositi kako na datoteku tako i na potkatalog. Potkatalozi su hibridi. Dodeljen im je prostor baš kao da su datoteke. Mogu da se povećavaju, šire i brišu kao datoteke. Ali umesto da sadrže naše podatke, oni sadrže imena drugih datoteka. Mogli bismo reći da je potkatalog po obliku nalik na svaku drugu datoteku, a da je po funkciji nalik na glavni katalog. Pošto su

Opis	Veličina (bajtovi)	Format
Ime datoteke	8	ASCII znaci
Proširenje	3	ASCII znaci
Atributi	1	Svaki bit predstavlja jedan atribut
Rezervisano	10	Neiskorišćeno (za sada)
Vreme	2	Reč kodirana
Datum	2	Reč kodirana
Početni FAT podatak	2	Reč
Veličina datoteke	4	Ceo broj

potkatalozi smešteni u područje podataka može im biti dodeljen prostor prema našim potrebama.

Svaka stavka u katalogu sadrži, između ostalog, ime datoteke, veličinu datoteke, datum i vreme pravljenja (ili poslednje izmene) datoteke i broj početnog klastera (FAT podatak) datoteke. Pogledajmo pobliže polje sa imenom datoteke u podatku direktorijuma.

Kao što možete očekivati, polje imena datoteke je područje od 11 bajtova podeljeno na polje imena od 8 bajtova i polje za proširenje (extension) od 3 bajta. Imena datoteka sa manje od 8 znakova dopunjuju se sa desne strane praznim mestima, odmah iza imena datoteke od 8 bajtova sledi polje za ekstenziju sa prostorom od 3 bajta (tačka nije upisana, podrazumeva se da se ona nalazi između osmog i devetog znaka). Ako koristite Directory Editor iz NU da biste pogledali ime datoteke, videćete da je ono uvek ispisano VELIKIM SLOVIMA, ako ste promenili ime datoteke u direktorijumu pomoću NU ili Directory Editor-a, uverite se da li ste upotrebili VELIKA slova, jer će DOS nagurati mala slova u ime datoteke.

Baš kao što FAT podatak može da bude i pokazivač za naredni podatak u lancu i jedan od tri posebna koda, tako i prvi bajt polja imena datoteke može da bude i prvo slovo imena datoteke i jedan od tri posebna koda.

Nula kao prvi bajt imena datoteke označava potpuno neiskorišćen podatak direktorijuma. Korišćenje 0 da bi se označile neiskorišćene stavke u katalogu omogućava DOS-u da zna kada je stigao do kraja aktivnih podataka u katalogu, a da ne pretražuje do kraja katalog (ova konvencija nije primenjena u DOS-u 1.X). Znak tačke kao prvi bajt imena datoteke kazuje da je DOS rezervisao taj podatak za plovidbu po strukturi kataloga.

Na kraju da pomenemo poslednje ali svakako ne i najmanje važno, a to je činjenica da prvi bajt imena datoteke može da sadrži i znak malog grčkog slova sigma. Ta posebna oznaka (čija je ASCII vrednost 229 dec, E5 hex) ukazuje da je datoteka izbrisana. Većina programa, uključujući UnErase i Quick UnErase programe, predstavljaju znak sigma kao znak pitanja da bi označili da on predstavlja neki izbrisan i sada nepoznat znak.

Srećom po nas i po UnErase program iz Norton Utilities a DOS je dosta lenj kada briše datoteke. Umesto da briše sadržaje podataka datoteke, on samo odstrani prvi bajt imena datoteke sa E5 da bi naznačio da je datoteka izbrisana, a zatim stavi nule u podatke o datoteci u FAT-u (setite se da vrednost 0 u FAT podatku govori da je klaster na raspolaganju). Pošto on nije stvarno izbrisao podatke, niti broj početnog klastera u katalogu, mi možemo vrlo lako da nađemo i ponovo oživimo prvi klaster pod uslovom da preko njega nije upisana neka druga datoteka.

Koristeći znanje o strukturi diska, direktorijuma i FAT-a, Quick UnErase program iz Norton Utilities često može automatski da spase ostatak klastera izbrisane datoteke, program čak zna koliko klastera treba da traži, pošto dužina izbrisane datoteke u direktorijumu nije izbrisana novim zapisom. Ponekad je datoteka tako loše izdeljena da program Quick UnErase ili ne može da nađe prave klastere ili nalazi prave klastere ali ih povezuje pogrešnim redom, to je vrsta problema koju program UnErase sa savršenijim sposobnostima treba da reši. Koristeći ono što znate o vašoj datoteci i ono što program UnErase zna o disku, često ćete moći da osposobite čak i loše izdeljene i delimično izbrisane datoteke.

Upotpunimo sliku

Do sada smo govorili o dva glavna područja na koje je disk podeljen — sistemsko područje i područje podataka — i o trima podgrupama sistemskog područja. Naime, rekli smo da je sistemsko područje podeljeno na početni zapis koji je uvek na prvom sektoru diska, FAT koji je odmah iza početnog zapisa i glavni katalog neposredno smešten iza FAT-a.

Ovaj opis sistemskog područja je stoprocentno tačan kod disketa i drugih specijalnih diskova koji ne mogu biti **podeljeni** da bi ih koristili višestruki operativni sistem.

Tvrdi diskovi visokog kapaciteta su vredna i skupa sredstva, i zato, kao pravi dobar komšija, DOS priskače u pomoć i nalazi način da podeli te diskove za razne operativne sisteme. Trik je u tome da se postavi više particija, po jedna particija za svaki operativni sistem (može se postaviti više particija koje sve rade u istom operativnom sistemu, ali malo je razloga da se to učini osim ako koristite verzije DOS-a do verzije 3.2 i osim ako imate disk veći od 32 megabajta, što je disk sa najvećim kapacitetom sa kojim te DOS verzije mogu da rade, a druge strane Compaq DOS 3.31 i PC DOS 4.0 mogu da podržavaju particije veće da 32 megabajta).

Pošto DOS ne može da uzme podatak o vremenu iz UNIX-ove particije, a UNIX ne zna kako da očita DOS-ove particije, sledi zaključak da mora postojati neko područje diska zajedničko za sve particije. Ovo zajedničko područje, poznato kao tabela particija (partition table), određuje mesto i dužinu na disku svake particije sistema.

Tabela particija se uvek postavlja u prvi sektor svakog diska deljivog na particije (kao što se i početni zapis smešta u prvi sektor svakog diska nedeljivog na particije). Na disku deljivom na particije tabela particija se mora postaviti pre logičkog formatizovanja (Možda se sećate da je logičko formatizovanje jedina vrsta formatizovanja koju na tvrdom disku DOS izvodi putem naredbe FORMAT ili programa Safe Format. Fizičko formatizovanje tvrdih diskova obično obavlja proizvođač). DOS-ova naredba FDISK postavlja tabelu particija! To je u stvari sve što čini. Dakle pre nego što upotrebite tvrdi disk prvo morate da izvedete FDISK, a zatim upotrebite FORMAT ili Safe Format.

Ako je disk deljiv na particije istovremeno i boot disk, iskrsava pitanje: kada ima više operativnih sistema na istom disku, kako PC može znati koji sistem da pokrene za vreme pultovanja? Odgovor glasi: zapis tekuće aktivne particije se takođe nalazi u tabeli particija.

Raznorazne kućne potrepštine se nalaze u tabeli particija, ali za nas je od istinske važnosti samo jedna. Rekli smo da na disku nedeljivom na particije prvi sektor uvek pripada početnom zapisu koji učitava ostatak DOS-a. Pošto DOS očekuje da nađe početni program u prvom sektoru, diskovi deljivi na particije takođe moraju da stavljaju početni zapis u prvi sektor kao deo tabele particija.

U slučaju kao što je ovaj, kada je početni zapis (početni program) deo tabele particija, on se naziva **matični početni blok** (master boot block). Sada, kada smo završili sa početnim programom sa našim diskom deljivim na particije, prvo damo učitati kod u matični početni blok tabele particija. Taj kod će naći koja je particija aktivna i onda će očitati početni kod (boot code) iz te particija. Tako da, ako DOS-ova particija zauzima ceo disk, matični početni kod (master boot code) učitati DOS-ov početni zapis, koji će zatim učitati ostatak DOS-a. Ako su tu dve particije i UNIX je aktivan, matični početni kod će učitati početni kod iz UNIX particije koji će, budući da je deo UNIX-a, učitati ostatak UNIX-a.

To je sve što se tiče tabele particija. Ostali deo diska deljivog na particije je organizovan kako je gore opisano, izuzev što se sada naš opis organizacije DOS diska odnosi samo na DOS particiju. U tipičnom slučaju, kada je ceo disk zauzet DOS particijom, jedine razlike bi bile te što se onda sve pomera za jednu stazu unapred da bi se napravilo mesta za tabelu particija (tabela particija koristi samo prvi sektor staze 0, ali preostali sektori na stazi se u svakom slučaju preskaču). DOS-ov početni zapis će i dalje počinjati u sektoru 1 ali na stazi 1, FAT će počinjati od sektora 2, staze 1 itd.

Kada smo kod FAT-a, hajde da još jednim detaljem zakomplikujemo stvari koje smo u dosadašnjem razmatranju bili pojednostavili. Setite se da smo govorili o FAT podacima od dva bajta, pošto dva bajta iznose 16 bitova i pošto 16 bitova može da primi tako veliki broj kao što je 65535, FAT podatak od dva bajta odgovarao bi diskovima koji sadrže do 65536 klastera.

Na žalost, FAT sa podacima od dva bajta je nalik na „kasno stiže Janko na Kosovo" (Johnny-come-lately), jer su u DOS verzijama do 3.0 FAT podaci bili od 1.5 bajtova. A 1.5 bajtova ili 12 bitova mogu da izraze brojeve samo do 4095, što je relativno mali broj klastera. Organizacija velikog tvrdog diska sa malim brojem klastera zahteva da svaki od njih ima veliki broj sektora, što, ako vaše datoteke nisu gargantuovske, nije efikasno.

Stoga, kao verzija 3.0, DOS prihvata FAT-ove sa podacima od dva bajta na velikom disku, dok i dalje koristi FAT sa podacima od 1.5 bajta na manjim diskovima. Bez obzira da li su podaci od 1.5 ili 2 bajta, FAT stalno radi na isti način, tako da je praktični značaj korišćenja FAT tabele sa dve različite veličine ravan nuli, ukoliko ne volite da radite FAT aritmetiku (što vam nije potrebno ako koristite FAT editor iz Norton Utilities Advanced Edition, on radi FAT aritmetiku za vas).

Ako koristite FAT editor, moram da pomenem još nešto: FAT je toliko važan da DOS od njega pravi dve identične kopije.

Na kraju, dozvolite da pomenemo još jednu sitnicu. Verovatno ste čuli za graničnu veličinu datoteke od 32 megabajta koju je nametnuo DOS i možda ste se upitali zašto je to tako. Pozabavimo se malo aritmetikom. Kao što se sećate, DOS transformiše fizičku trodimenzionalnu šemu numerisanja (strana/staza/sektor) koju koriste disk kontroler i BIOS u jedan sekvencijalni jednodimenzionalni broj sektora. DOS koristi reč od dva bajta da bi odredio taj broj sektora, tako da najveći broj sektora kojim može da barata iznosi 65535 — to je najveći broj koji se može izraziti u dva bajta.

Ako imamo 65536 sektora (DOS-ovo logičko numerisanje počinje od 0 bez obzira što fizičko numerisanje sektora počinje od 1) a svaki sektor ima 512 bajtova onda je maksimalan broj bajtova koje možemo adresirati 65536 puta 512 što je 32 megabajta

Način da se pređe granica od 32 megabajta je da se poveća veličina sektora Sa veličinom sektora do na primer 1024 mogli bismo da adresiramo do 64 megabajta To je put kojim su krenuli prodavci nekih velikih tvrdih diskova koji se sada nude za PC Compaq-ov DOS 3 3 1 je prvi prevazišao to ograničenje povećavajući FAT podatke od dva bajta na četiri bajta Ranije je podatak od dva bajta mogao da označava do 65536 sektora Međutim koristeći četiri bajta, ta brojka se povećava na 4 milijarde sektora (na kog astronoma vas to podseća?) Stoga ova DOS-ova verzija može da operiše sa particijama većim od 32 megabajta PC-DOS i MS-DOS 4 0 takođe podržavaju particije veće od 32 megabajta koristeći tu istu šemu

Stvari koje se iznenada gube na disku

Sada kada znamo kako disk treba da radi prilično dobro možemo da shvatimo i šta može da krene pogrešno i — što je još važnije — kako da to izbegnemo ili popravimo

Dosta je lako zamisliti šta može da krene naopako između kataloga i FAT a Moguće je na primer da u FAT u neki klaster bude označen kao da je u upotrebi a da ipak ne bude deo lanca podataka koji se odnose na bilo koju datoteku Takva anomalija bi ukazivala na postojanje nekog napuštenog klastera (koji se ponekad naziva izgubljenim klasterom — lost



cluster) Budimo još maštovitiji zamislimo da nas FAT podatak za neki klaster umesto da nas uputi dalje vraća upravo na onaj podatak koji nas je i doveo do njega stvarajući tako kružni lanac Takođe se može desiti da dva ili više podataka ukažu na isti klaster što znači da je jedan isti klaster dodeljen dvema različitim datotekama To je greška poznata kao ukrštene datoteka (cross linked file)

Ovakva greška se mogu dogoditi ili kada je pun mesec i kada noću lutaju čudne spodobe ili kada isključite svoj računar dok program još ima otvorenu datoteku ili još upisuje na disk Izvedite svoje programe do kraja pre nego što isključite mašinu

U svakom slučaju *Norton Disk Doctor* utility može otkriti takve logičke greške saopštiti ih i popraviti i to uz minimalnu intervenciju korisnika U tom pogledu je mnogo savršeniji od programa DOS CHKDSK Naime CHKDSK će vas zdravo obmanuti ako ga pokrenete bez /F Reći će vam da je ispravio greške ali nemojte mu verovati

Štaviše CHKDSK ne ispituje i ne može da popravi fizičke greške na disku kao što su sektori koji se ne mogu očitati Disk Doctor može da reši tu vrstu problema Program može ponovo da formatizuje loše sektore i ponovo upiše stare podatke Osim toga Disk Doctor može da otkrije i ispravi loše tabele particija da otkrije i popravi loše početne zapise (ponovo osposobljavajući disk za butovanje) da otkrije i otkloni probleme pomoću FAT a i proveri i ispravi glavni direktorijum

Disk Doctor na neki način predstavlja pretakanje znanja sadržanog u ovoj brošuri u vrlo moćan program Disk Doctor je zamišljen tako da čak i relativnom početniku daje moć da reši raznorazne probleme sa minimumom napora Međutim sve i da vam ne treba da znate kako vaši diskovi rade da bi ste ih održavali u najboljem stanju nešto više informacija nikome ni kada ne škodi izuzev možda nekolicini nesrećnika u Volstritu

Ako ste došli dovde onda mora da ste dosta dobro shvatili šta se dešava iza onog malog crvenog svetla na disku Nismo naravno razmotrili sve što treba znati o disku ali smo obuhvatili sve važne pojmove i izraze sa kojima ćete se verovatno sresti u svakodnevnom radu sa računarom i DOS-om

Ako odlučite da proverite nešto od onoga što ste naučili da istražujete svoj disk i možda čak nađete neke stvari koje ovde nismo imali mesta da pomenemo otkrićete da su programi *Norton Utilities* kako zgodni za istraživanje vašeg diska tako i korisni za obnavljanje podataka

ZA PRETPLATNIKE SPECIJALNI POPUST OD 20%

VAŠ NAJBOLJI VODIČ KROZ SVET RAČUNARSTVA

računari

časopis za informatiku i računarstvo

Svakog prvog u mesecu na najbližem kiosku
ili **u vašem poštanskom sandučetu**

Pretplata vam donosi nekoliko važnih prednosti:

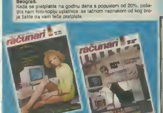
— **POPUST OD 20%:** umesto za 120 000 D 12 brojeva dobijate za 96 000 D

— **GARANTOVANU CENU:** barem što se „Računare" tiče, neprijatna i neizbežna poskupljenja će prestati da vas brinu

— **BRZU ISPORUKU:** po pravilu, svoj časopis ćete dobijeti pre nego što se pojavi na kiosku

— **JEDNU BRIGU MANJE:** ubuduće umesto da vi mislite na „Računare" oni će misliti na vas!

Da biste dobili odgovarajuću uplatnicu za pretplatu obratite nam se na adresu **RAČUNARI, Bulevar vojvode Mišića 17, 11000 Beograd.**
Kada se pretplatite na godinu dana s popustom od 20%, pošaljite nam foto-kopiju uplatnice sa tačnom naznakom od kog broja želite da vam teče pretplata



računari
su uvek aktuelni

Ono što nema u novom, potražite u starim brojevima

- novi hardverski uređaji, testovi, uporedni prikazi
- komercijalni softver, softverski podsetnici testovi
- najnovije vesti sa domaćeg i stranog tržišta
- tehnike programiranja, algoritmi, biblioteke programa
- klub programera, help, bajtovi lične prirode
- računari u školi računari u maloj privredi

MOGU SE NARUČITI SVI BROJEVI OD 40 DO POSLEDNJEG (BROJEVI OD 1 DO 39 SU RASPRODATI)

CENA PO PRIMERKU

računari 40 2 000 D	računari 46 3 500 D
računari 41 2 000 D	računari 47 3 500 D
računari 42 2 000 D	računari 48 5 000 D
računari 43 2 000 D	računari 49 8 000 D
računari 44 3 000 D	računari 50 8,000 D
računari 45 3 500 D	računari 51 8 000 D

POŠTARINA

Za pošiljku se — na ime troškova pakovanja rukovanja i poštarina — plaća 2 000 plus 500 D po naručenom primerku „Računara" Tako na primer, za naručenih šest brojeva plaća se 2 000 plus 3 000 ukupno 5 000 dinara

POSTUPAK NARUČIVANJA

Da bi se pojednostavila procedura, smanjili troškovi za kupca i ubrzala isporuka molimo da se pridržavate sledećih pravila

1) Izaberite brojeve „Računara" koje želite da kupite saberite njihova cena iz pregleda koji objavljujemo i na dobijenu sumu dodajte još 2 000 D za pošiljku plus 500 D za svaki naručeni primerak

2) Ukupni iznos uplatite na žiro-račun 60802-603-23264 (RO BIGZ Bulevar vojvode Mišića 17, 11000 Beograd) sa obaveznom naznakom da je reč o kupovini starih brojeva „Računara" (navedite brojeve)

3) Potvrdu o uplati ili njenu foto-kopiju pošaljite na adresu redakcije Računara" (kao pod 2) Redakcija se obavezuje da će vam naručeni brojevi biti upućeni najkasnije deset dana nakon prijema potvrde o uplati Za eventualne reklamacije javite se Prodajnom odeljenju BIGZ-a (011/650-582) ili redakciji (011/653-748)

Đaci, nastavnici, studenti, asistenti!!!

NAJJEFTINIJI PRIRUČNIK ZA NASTAVU I VANNASTAVNE ŠKOLSKE AKTIVNOSTI IZ INFORMATIKE I RAČUNARSTVA

Ako ste propustili
još uvek možete da nabavite
„Računare" broj 48 (mart 1989)
sa specijalnim dodatkom
na 32 strane velikog formata

RAČUNARI U ŠKOLI (broj 2)

- PROGRAMERSKA VEŽBANKA
- STRUKTURIRANI BEJZIK
- MOJ PRVI ALGORITAM
- INFORMATIKA I DRUGI PREDMETI
- VODIČ ZA VEŽBANJE

„Računari" broj 48 po ceni od 8 000 D po primerku mogu se naručiti na adresu **RAČUNARI BIGZ, Bulevar vojvode Mišića 17, 11000 Beograd, ili na telefone 011/650-528 (prodaja), 011/653-748 (redakcija)**
Školama bibliotekama đačkim zadrugama odobravamo rabat od 20% (važi samo kod kupovine više od pet primeraka)

Problem svemirskih hirurga

Iako u literaturi relativno dobro poznat, problem svemirskih hirurga je izazvao pristojan odziv (106 pisama) i doneo čitavih 12 pogrešnih rešenja!

Uređuje Dejan Ristanović

Podsetimo se, pre svega, ove humanitarne Pitalice. Svemirski brod „USS Enterprajz" dobio je zadatak da obiđe četiri planete Federacije i prevaze četiri slavna hirurga specijaliste za razne tipove operacija na Zemlju kako bi uzeli učešće u radu značajnog konzilijuma. Svaki od hirurga pripadao je drugoj rasi — iako su njihova pojava bile slične ljudskima, biohemija njihovih organizama bile su suštinski različite tako da bi i najmanji kontakt između bilo koja dvojice doveo do trenutne smrti. U početku je sve bilo u redu, ali je nedaleko od Zemlje jedan od hirurga dobio težak napad Kalimato sindroma praćen teškim unutrašnjim krvarenjem. Život mu je bio ozbiljno ugrožen, pa su trojica njegovih kolega održala hitan sastanak i zaključila da brza hirurška intervencija pruža značajne izglede za spas. Pokazalo se da je operacija veoma složena i da zahteva da sva tri hirurga sekvencijalno (prvi, drugi pa treći) obave deo koji im je specijalnost. Nesreća, međutim, nikada ne dolazi sama — pripremajući se za operaciju hirurzi su primetili da „Enterprajzova" bolnica uopšte nije spremna za operativne zahvate: nisu pronađena čak ni tri para hirurških rukavica!

Da operaciju obavljaju ljudi, moglo bi se zamisliti da dva lekara upotrebe iste rukavice. Za svemirske hirurge je, međutim, slično rešenja bilo nemoguće — ukoliko bi neko navukao rukavice koje je nosio njegov prethodnik, došlo bi do indirektnog kontakta njihove kože i do trenutne smrti. Nije se, naravno, mogao zamisliti ni rad bez rukavica jer bi direktan kontakt između hirurga i pacijenta (obojica sa različitih planeta) bio fatalan za obojicu. Hirurzi su ipak bili dovoljno domišljati i obavili posao sa svega dva para rukavica. Trebalo je pogoditi kako.

Prvi hirurg je najpre navukao jedne rukavice i onda preko njih druge i obavio svoj deo operacije; time je „kontaminirao" spoljašnju stranu spoljašnjih rukavica i unutrašnju stranu unutrašnjih. Onda je drugi hirurg navukao samo spoljašnje rukavice (čija je unutrašnja strana čista) i obavio svoj deo operacije — spoljašnja kontaminirana strana je ponovo bila u dodiru sa pacijentom ali to prema uslovima zadatka ne proizvodi nikakve posledice. Najzad je treći hirurg obukao **prevrnute** unutrašnje rukavice (prevrtanjem, naravno, leva rukavica postaje desna i obratno) i preko njih spoljne rukavice — kontaminirana spoljna strana unutrašnjih rukavica došla je u dodir sa (takođe kontaminiranom) unutrašnjom stranom spoljnih rukavica, ali je pacijent i dalje bio zaštićen time što je u dodir sa njim dolazila samo ona strana rukavica koja je započela operaciju. Pretpostavka je, naravno, da se na brodu nalazi neko (npr. član posade) ko može da prevrće rukavice i navlači ih hirurzima na ruke.

Zanimljivo je da je nekoliko rešavača pronašlo i alternativno rešenje koje se u literaturi ne spominje: prvi hirurg stavlja dve rukavice na levu i jednu na desnu ruku, drugi hirurg na jednu ruku stavi neupotrebljavanu rukavicu i preko nje spoljnu rukavicu sa prve ruke prvog hirurga, a na drugu ruku prevrnutu donju rukavicu sa prve ruke prvog hirurga i preko nje rukavicu koju je prvi hirurg imao na drugoj ruci i tako dalje. Drugi potencijalni prilog svetskoj literaturi je formula po kojoj N hirurga sa raznih svetova može da obavi operaciju primenom svega INT (N/2) + 1 parova rukavica.

Kompjuterskih rešenja je, prema očekivanju, bilo malo, ali ih zato nagrade nisu mimoišle — 120.000 dinara dobiće *Slavko Kukika* iz Ljubljane koji je uz pomoć IBM PC AT a i programa pisanog na *Turbo Pascal*-u za manje od sekunde dobio kako osnovno tako i alternativno rešenje, 80.000 dinara *Davorinu Ruševljanu* iz Zagreba koji je zadatak rešio na Prologu (Atari ST je radio oko 15 minuta) a 40.000 dinara *Nenadu Radulovicu* iz Kutine koji je imao najviše sreće u improvizovanom izvlačenju. Posebne pohvale zaslužili su *Zoran Jovičić*, *Zoran Rogić* i *Jan Slivka*. Svi nagrađeni programi su na žalost predugački da bismo ih ovde publikovali.

Zanimljivo je dodati da ovaj zadatak ima i malo drugačiju (po prilici stvari mnogo logičniju) postavku u kojoj su tri ne baš zdrava mornara posle dužeg putovanja posetili luku i jednu njenu (takođe bolesnu) stanovnicu. Postavku smo, međutim, procenili neprikladnom za objavljivanje u „Računarima" ali ipak vrednom ovog malog spomena.

Tajanstveni broj

Broj X je proizvod četiri različita prosta broja među kojima se ne nalazi jedinica. Kvadrat broja X sastoji se od 9 cifara, pri čemu su prve tri iste kao i zadnje tri, dok su srednje tri cifre jednake zbiru brojeva koji nastaju od prve tri i od zadnje tri cifre. Ukoliko je kvadrat broja X broj AAABBBCCC biće AAA=CCC i BBB=AAA+CCC=2*AAA. Vaš zadatak je da pronađete najveći mogući broj X koji zadovoljava ove uslove.

Rešenja Pitalice, zajedno sa programima koje ste koristili da ih dobijete, pošaljite na uobičajenu adresu „Računari" (za Dejanove pitalice), Bulevar vojvode Mišića 17, Beograd tako da pristignu pre 25. jula 1989. Sva pisma sa korektnim rešenjima konkurišu za novčane nagrade od 120.000, 80.000 i 40.000 dinara, dok će kuponi (ili njihove fotokopije) na koje je upisan identifikacioni broj učestvovati u godišnjem takmičenju rešavača Pitalica. Identifikacioni broj dobijate tako što u prve tri kućice upišete **poslednje** tri cifre nečijeg broja telefona, u sledeće dve godinu vašeg rođenja i na kraj dodate dve cifre po izboru. Obratite pažnju da sva rešenja koja šaljete u toku godine nose isti identifikacioni broj.

Q zadatak

RQ18. U centru Londona postoji jedno pozorište koje s vremena na vreme menja ime. Pre 1901. godine ovo se pozorište, na primer, zvalo A, onda je „prekršteno" u B da bi 1952. ponovo dobilo ime A. Gotovo je sigurno da će u budućnosti pozorište ponovo dobiti ime B. Trenutno ime tajanstvenog pozorišta je **Queen's Theatre** i važi od trenutka kada je na Britanski presto stupila Elizabeta II. Od 1901. (kada je Eduard VII stupio na presto) pozorište se zvalo **King's Theatre** a pre toga opet **Queen's Theatre**, po kraljici Viktoriji.

RQ19. Suprotno ustaljenom običaju, imali smo jednu programersku Q Pitalicu iz stvarnog života. Nedavno smo pokušavali da ugradimo PC karakter set u jedan inače Epson FX 80 kompatibilan matrični štampač: pronašli smo pozicije svih italik znakova, preko njih presnimili odgovarajući segment ROM-a Epsona FX 85 (PC karakteri), programirali EPROM, poslali sekvencu Esc 4 (proširivanje opsega znakova koji se ispisuju) i testirali štampač. Sve je bilo u redu osim što je umesto slova DJ (kod 178 u standardu „Računari") ispisivan blanko. Zaista nije bilo lako pogoditi zašto. Epson kompatibilni matrični štampači u ROM-u čuvaju dva oblika nula — običnu čiji je kod 48 (176 ako se radi o italic nuli) i precrtanu čiji je kod 127 (255 za italic). Od položaja jednog mikroprekidača zavisi koja će se nula ispisivati — naš mikroprekidač je bio u položaju „precrtana nula" pa je štampač umesto koda 176 ispisivao kod 255 koji je u Epson/PC karakter setu blanko! Kada jednom shvatite u čemu je problem, do rešenja se lako dolazi — treba samo upisati oblik slova DJ u ćelije koje zauzima CHR$(255).

RQ20. Srpskohrvatske reči koja čitana sleva nadesno ima slovo K a čitana sdesna ulevo nema K dosetio se, kažu, još Vuk Karadžić — reč KAMEN.

Q21. Prvi zakon robotike, prema idejama Isaaca Asimova, kaže da robot ne sme povrediti ljudsko biće niti dopustiti da njegovom pasivnošću ljudsko biće bude povređeno; ovaj se zakon ugrađuje u samu suštinu pozitronskog mozga tako da ga robot nikako ne može prekršiti. Zabeležen je, međutim, slučaj u kome je robot ubio svog servisera. Kako je to moguće (prema ideji *Eduarda Grubera* iz Novog Sada)?

45. Najveći broj X je ____________

Ime i prezime ____________

Adresa ____________

Mesto ____________

Kompjuter ____________ Vreme ____________

Identifikacioni broj: |_|_|_| |_|_| |_|_|

iz broja tel. — god. rođ. — po izboru

132 [KOMS NORTON, KDMS PCTOOLS PC] Iako bombastio najavljen i agresivno reklamiran Nortonov disk doktor (NDD iz *Norton Utilities Advanced Edition 4.50)* nije baš uvek spasonosan. Nedavno nam je ovaj program spasao podatke sa jedne na nepoznat način „sludane" AT diskete ali nas je uvalio u veliku brigu.

Pored ostalih problema NDD je prijavio da postoji fizička greška u FAT-u diskete. Greška je ispravljena, podaci prepisani na disk, disketa formatirana i sve vraćano na nju. Ponovo smo startovali NDD i iako starih problema nije bilo pročitali da je FAT i dalje fizički oštećen. Pomislili smo da je disketa neispravna i probali drugu, treću, četvrtu... svaka AT disketa je imala fizički loš FAT dok je na svim 360 K, 720 K i 1.44 M disketama bilo sve u redu. Startovali smo sistem sa DOS diskete, ponovili NDD i sve AT diskete su bile ispravne.

Šta čovek da pomisli u ovoj situaciji? Naravno da je sistem zaražen virusom i da se radi o nekom njegovom artefaktu. Posle dobrih pola sata analize zaključili smo da je problem mnogo manje ozbiljan — NDD se ne slaže sa PC CACHE-om (PC-CACHE /SIZEXP=xxxK na *Protec* ploči čija je *Extended* memorija odgovarajućim drajverom pretvorena u *Expanded)*. Zaključak: pre nego što startujete NDD, isključite keš.

133 [INT 6.33, PC, ST] Posle tačno devet meseci i tačno sto bajtova lične prirode vraćamo se problemu učitavanja Atari disketa na PC-ju. Metod opisan u „Računarima 43" (NU/M promena prva tri bajta *boot* sektora u &EB &34 i &90 i promena prvog (tj. nultog) bajta FAT tablice (staza 0, sektor 2) sa &F7 na &F9) radi i ne radi — nekada NU odbije da opšti sa disketom, nekada se *root* katalog gubi, nekada se direktorijumi pretvaraju u datoteke.

Problem se može razbiti na dve komponente: praktičnu i apsolutnu. Praktična strana zahteva da ostvarimo prenos „atari" ↔ PC i potpuno je rešena: disketu formatiramo **na PC-ju** (ili, ako je to nemoguće, na „atariju" koji radi pod *PC Ditto-m)* i na preveći nikakve direktorijume na njoj, datoteke koje se prenose upišemo u osnovni *(root)* katalog. Pretpostavka je, naravno, da je „atari" koji koristimo opremljen dvostranom disk jedinicom — PC jednostavno ne poznaje jednostrane diskete od 3.5 inča!

Potpuno rešenje bi obuhvatalo odgovor na pitanje *zašto NU ne može da pristupi nekim Atarijevim disketama* i program koji čita sva ku Atari disketu (pa i jednostranu) sa posebnim osvrtom na diskete koje su „imunizovane" pomoću neke vakcine koja muči PC-ja. Ovakvim rešenjem na žalost još ne raspolažemo. A vi?

134 [OS MSDOS, PC] Ako volite da testirate ispravnost upisa na disketu i još koristite neki keš program, očekujte poneko iznenađenje. Nedavno smo na primer kopirali neke važne datoteke na disketu od 360 K i poželeli da sa COPY A:*.* NUL (ili što je uglavnom isto Nortonovim DT A: /F) proverimo da li je sve došlo na svoje mesto. Lampica na disku se jedva upalila i PC je prijavio da je sve u redu — nije naravno čitao baš ništa jer je svih instotinak kilobajta podataka počivalo u kešu!

Mogućih rešenja ima više. Na primer otkucali PC-CACHE /FLUSH (ako vam je ovo često potrebno možete da formirate FLUSH.BAT datoteku odgovarajućeg sadržaja). Ili koristili opcije /IA odnosno /IB pri instaliranju PC-CACHE-a — time se ukida beleženje „saobraćaja" sa flopi diskovima u keš. Ili što dolazi u obzir samo ako hardver obaveštava operativni sistem da je disketa vađena (većina AT računara) otvori ti klapnu, izvaditi disketu, vratiti je nazad i tek onda počeli kopiranje „u ništa".

135 [DTXT VENTURA 1.13, PC] U „Računarima 50" smo videli da pri konverziji *portrait* strnice u *landscape* format *Ventura* automatski konvertuje IMG slike u IML posle čega je korišćenje tih slika moguće i u drugim *portrait* tekstovima, zgodno kada neku sliku treba „prevrnuti". Da bi se međutim slika stvarno iskoristila treba obrisati (ili preimenovati) originalni IMG fajl i onda izvršiti RENAME IME.IML IME.IMG. [Prilog **Zoran Životić**].

136 [DTXT, VENTURA, PC] Ostanimo još malo kod IMG fajlova. Kada naložite *Venturi* da učita neku PCX ili TIF sliku, ona će biti automatski konvertovana u istoimeni IMG, jedini format sa kojim *Ventura* operiše. Ukoliko docnije napustite *Venturu* i promenite originalnu (PCX ili TIF) sliku, izgled *Ventura* dokumenta se neće promeniti — *Ventura* ne zna da je slika promenjena pa koristi stari IMG fajl. U nedostatku MAKE uslužnog programa koji bi proverio vajući datume otkrio koje je slike starije, pre ulaska u *Venturu* obrišite sve IMG fajlove koji odgovaraju promenjenim slikama.

137 [KDMS NORTON, PC] Jeste li znali da pozivu raznih programa iz paketa *Norton Utilities* možete da dodate opciju /T i tako posmatrate zbir a ne komponente — FS *.* /S /T na primer daje zbirnu dužinu svih datoteka u tekućem katalogu i njegovim potkatalozime bez ispisivanja kilometarske liste imena datoteka.

138 [DTXT WPERF, PC] Znate li kako se zaglavljuje *WordPerfect 5.0*? Otkucate Alt 240 i čekate, čekate, čekate... Ponekad će pomoći da pritisnete nekoliko tastera ili Ctrl Break a nekada ćete morati da resetujete kompjuter! [Prilog **Dragan Mitraković**]

139 [PJ PASCAL 1.21, PC] Osim mogućnosti korišćenja memorije na *heap-u* metod opisan u prošlim „Računarima" nudi još po neko unapređenje, može se na primer, alocirati onoliko memorije koliko je stvarno potrebno! Pretpostavimo da smo deklarisali tipove *redt = array [1..60000] of byte* i *redpt = ^redt* a zatim i promenljivu *red: redpt*; ostaje nam samo da izvršimo *getmem (red, sizeof (redt))* i tako alociramo memoriju za ovaj gigantski red. Ukoliko međutim program koji se izvršava po učitavanju ulaznih podataka zaključi da će red biti dug svega 20000 znakova, nema prepreke da se izvrši *getmem (red, 20000)* ili predostrožnosti radi *getmem (red, 25000)*. Kompajler jasno nema načina da zna koliko će memorije biti alocirano za neki niz što znači da posle ovakvih egzibicija *Range Check On* nema preveliku vrednost — ukoliko usled bega u programu, referenciramo hiljaditi bajt niza, čitaćemo ili (još gore) pisati o segmentu memorije koji je prazan ili, prema Marfijevim zakonima, zauzet nekim drugim podacima.

140 [ZAST, SHARP] Većina Sharp-ovih džepnih kompjutera (npr. PC—1401, PC—1402 itd) ima komandu PASS kojom se može „zaključati" važan program i tako sprečiti njegova promena ili kopiranje. Šta međutim da radite kada zaboravite šifru ili kada vam neki „prijatelj" zaključa kalkulator? Ništa lakše: sedam bajtova počevši od 1&113 krije ASCII reprezentaciju šifre — CHR$(PEEK(1&113))+CHR$(PEEK(1&114))+ ... će je ispisati na ekranu! [Prilog **Dejan Grujić**, Pančevo]

U sledećim „Računarima" objavićemo kako se „otključavaju" PRIVATE programi na HP—41C.

141 [MREZ, PC] Imate li modem? Ako imate, i jugoslovenska kompjutersko-komunikaciona scena ima svetlih tačaka u vidu Mail Box servisa koji omogućavaju slanje i prijem poruka, učešće na hakerskim konferencijama, razmenu programa, oglašavanje... U očekivanju opširnijeg teksta koji bi se bavio YU računarskim komunikacijama objavljujemo listu trenutno aktivnih servisa sa telefonima, radnim vremenom i modemskim parametrima. Listu smo (posredstvom modema, naravno) dobili od niškog kluba VIK — ako su nekom čitaocu ove rubrike poznati Mail Box klubovi koji nisu na spisku, pozivamo ga da nam se javi.

YU MailBox Servisi

Ime	Telefon	Brzina	Parametri	Radno vreme
Atari BBS	011 259 205	300-1200	8/N/1	22 do 08 h
SUMMA BBS	041 275 039	300-2400	8/N/1	18 do 08 h svaki dan
			7/E/1	Non-Stop vikendom
MIPS BBS	041 621 657	300-2400	8/N/1	Non-Stop
SOS BBS	097 22 580	300	8/N/1	22 do 08 h
VIK MBX	018 88 073	300-2400	8/N/1	21 do 02 h
Zagreb BBS	041 535 349	300-1200	8/N/1	Nonstop
			7/E/1	

142 [IGRE, PC] Igrate li SOKOBAN? Radi se o logičkoj igri koja u krugovima kompjuterskih profesionalaca teži da zameni TETRIS — sve veća popularnost u našim institutima i laboratorijama kao da implicira novu igračku groznicu!

Poznavaoci *Boulder Dash*-a i *Repton*-a bi možda rekli da SOKOBAN i nije nešto posebno ili novo ali kada na PC-ju nema ničeg boljeg...

Priloge za ovu rubriku šaljite na adresu: Računari (za Bajtove lične prirode), Bulevar vojvode Mišića 17, Beograd



HELP

Priprema Žarko Berberski

U avom sedamnaastom izdanju rubrika „HELP“ ja u znaku javnog softvara. Zbog rastućeg broja programa, komprimovanih fajlova i broja diskata koja stižu u redakciju, uvedan ja novi sistam snimanja koji bitno ubrzava i olakšava čitavu operaciju. Porad toga, od ovog broja HELP kroz saradnju sa programarima iz Holandija obezbaduja PD magazin ST_NEWS koji se diatribuira na diakatama po čitavoj Evropi. Takođa ća avi domaći PD programi aa porukama na anglaskom biti posiati u Holandiju, odakla će sa dalja distribuirati po Evropi.

Sretno Programiranje

PC/EPSON

Promena skupa znakova

Krešimir Majdenić iz Donjeg Milanovca ima probleme sa promenom skupa znakova na PC kompatibilcu i epson RX-100 štampaču

Za Krešimira stižu samo crne vesti. Hercules kartica ne dozvoljava softversko menjanje skupa znakova, već se mora menjati ROM, što će verovatno umeti da uredi svaki bolji servis. Ukoliko se odlučite za ovo rešenje, izaberite raspored koji odgovara Računar-standardu jer će vam YUSCII (koji je sedmobitan) uništiti mnogo bitnih znakova (zagrade, majmun itd.). Sve posebne znakove koje imate nameru sami da dizajnirate stavite takođe u višu polovinu tabele (128–255). Pri tom pazite da ne uništite neko od znakova koji vam treba (pri kraju tabele postoji dvadesetak matematičkih simbola koji su za vaš problem vrlo bitni). Ukoliko vam je softverska promena znakova baš toliko bitna, moraćete da kupite bar EGA karticu ili da se naoružate sa dosta strpljenja i pređete u grafički mod. Ukoliko se na kraju ipak odlučite za rad u grafičkom modu (štampanje i dalje možete izvoditi u normalnom — „karakterskom“ modu) bilo bi dobro da se opredelite za neki od već postojećih grafičkih sistema (GEM, MS Windows).

Štampač „epson RX-100“ je jedan od onih modela koji ne dozvoljavaju nikakvo softversko definisanje znakova, pa vam ovde ostaju dve alternative: lošija — zamenite ROM i bolja — prodajte štampač i kupite novi i to darem NL-10 odnosno LC-10. Pazite da ne uzmete „epson LX-800“ jer on omogućava definisanje svega nekoliko znakova što nije dovoljno ni za sve naše latinične znakove a kamoli za vaše matematičke simbole. Inače i ovde važi sve ono što je o izboru standarda i rasporedu dodatnih znakova rečeno za Hercules karticu.

STAR LC 10

LC-10 i WordPerfect

Nemeth Atila iz Novog Sada ima štampač „star LC 10“ koji ga „ne sluša“ kad radi sa WordPerfect-om

Za pomeranje papira nekoliko linija u napred najverovatnije je kriv „drajvar“ ili format stranice koji ste zadali. Prazne zaglavlja i druži na koja ne odgovara dužini vašeg papira mogu učiniti pravi haos prilikom štampanja. Star NX-1000 je u teoriji potpuno softverski jednak sa LC-10, no za svaki slučaj prekontrolišite sve kodove a pre toga svakako proučite deo uputstva za WordPerfect koji govori o tome i sve vreme konsultujte uputstvo za štampač (za koje pretpostavljam da ste ga već proučili). Jedan od kritičnih momenata je takozvana inicijalizacija stranice, tj. niz kodova koji se šalju pre štampanja svake stranice.

Što se domaćih znakova tiče, koristite mogućnost za „download“ koju i WordPerfect podržava, omogućavajući vam da pre štampanja dokumenta pošaljete štampaču čitav laj pun kontrolnih kodova. Pazite naravno raspored kodova. Veći deo onoga što stoji u odgovoru Majdenić Krešimiru u ovom istom Help-u važi i za vas. Pročitajte osim toga tekst Dejana Ristanovića u RA 42/40. Tekst nije za vaš štampač ali će sasvim sigurno mnogo pomoći.

PD igre za toplo leto

Prvi deo letnjeg paketa igara dobijate u ovom broju, a drugi ide u sledećem. U prvom paketu su tri foldera. Prvi je HUNGRY, igra do-

Horvat odgovara Caru

U RA 48/80 HELP je objavio poziv Boždara Horvata za pomoć kao i njegov protest zbog nekorektnog poslovanja firme „Tehničar“ i njenom odnosu prema kupcima u procesu eksploatacije računara. U RA 50/80 objavili smo integralni odgovor direktora „Tehničara“ Branka Cara. Sada je pred vama i **integralni tekst** odgovore Boždara Horvata.

Po objavljivanju „Terinog“ odgovora iz Zagreba nam se telefonom javio čitalac iz jedne druge radne organizacije koja ima sličnih problema. U razgovoru je postalo jasnije da je bitan problem (pored toga što „Tehničar“ drži mašinu „zatvorenom“) u formatu zapisa na disku. Pošto ovaj format u većoj meri diktira kontrolerski čip, moglo bi se pokušati sa povezivanjem neke od standardnih disk jedinica i nešto eksperimentisanja. Druga i mnogo perspektivnija varijanta je povezivanje serijskom vezom sa „amstrad CPC 6128“ mašinom koja je najrasprostranjenija mašina sa CP/M-om u nas. Ukoliko bi ova veza proradila, verovatno bi svi problemi bili rešeni jer je CP/M potpuno standardizovan sistem pa se programi s jedne mašine lako izvršavaju na drugim mašinama. Moglo bi se osim toga razmišljati i o povezivanju sa „im“ mašinama za koje je sve više i više softvera. Za sve ovo je potrebno da „Tehničar“ makar malo otvori svoju mašinu.

Poštovani drugovi,

najljepše Vas molim za dio dragocjenog prostora u „Računarima“. Naime, želio bih u nekoliko rečenica odgovoriti drugu Branku Caru, direktoru „Tehničara“, kako njegova profesionalna slatkorječivost ne bi u javnosti stvorila krivi dojam (kao da je svoju reakciju pretvorio u EPP poruku).

Evo teksta:

U aprilskim „Računarima“ objavljeni su izvodi iz mojeg pisma kojem je prvenstvena namjera bila da nakon bezuspješnog traganja odgovore na svoja pitanja pokušam dobiti od iskusnijih kolega iz „Računara“ (Na identično pismo jednom drugom časopisu za kompjutere dobio sam odgovor koji počinje „Za računalo Tera prvi put čujemo pa...“)

Napominjem da sam u pismu ***obrazložio*** *svoja* ***iskustva****. Dakle, namjera mi nije bila da neargumentirano napadam Taru, već sam jednostavno iznio svoja na žalost neugodna iskustva koja sam stekao u radu s „Terom 3“ i kontaktu sa RO „Tera“.*

Mislim da je nepotrebno komentirati CP/M na koji je instaliran spomenuti sistem, važnije je još jednom upitati se što može nekoga motivirati da u današnje vrijeme konstruira potpuno nestandardni format diska.

Mislim da se u takvom poslovanju prodaja proizvoda po sistemu „ključ u ruke“ (kako reče drug Car) taj izraz može vrlo lako promijeniti u „ključ na ruke“. Nadalje drug Car se pita na osnovu čega zaključujem da je njihov software loše kvalitete.

Jednostavno promatrajući današnje moderne aplikacije i uspoređujući ih sa spomenutim (o cijenama — bolje neću). Kako netko može objasniti „nevjerojatnu“ sposobnost nekih programa da bez ikakvog razloga usred unosa nekog podatka jednostavno zablokira sistem čime je naravno uništen dugotrajan i mukotrpan rad.

Posebno je interesantna periodičnost tih pojava. Za vrijeme obavljanja prakse pokušao sam više puta obratiti se drugovima iz „Tere“.

Svoja iskustva zaokružio sam nakon telefonskog razgovora s jednim od drugova, koji je na moje traženje nekog od uslužnih programa ili programskog jezika rekao „da može razgovarati samo na višem nivou tj. s nekim od direktora“, zatim „da njima nije u interesu da prodaju programske jezike“. Teško je povjerovati da se to događa u današnje vrijeme modernih kompatibilnih PC računala.

Još bi se dalo pričati o Terinim dodatnim kursevima (naši operateri su donijeli nekoliko naživrjanih CP/M naredbi) kao i o „detaljnim“ uputama uz software.

Međutim to već prelazi dobrim dijelom u domenu odgovornosti same RO koja je navedeno tražila.

Na kraju bih se želio izviniti što iz razumljivih razloga ne mogu spomenuti ime buduće (?) RO bez prethodne suglasnosti rukovodioca.

Žao mi je ako je mojim komentarom ugrožena reputacija „Tehničara“ (koja je ipak najčvršća na polju stolnih kalkulatora, pisaćih strojeva i sl.), pogotovo u slučaju ako je moja RO iznimka, a u što bi se vrlo rado uvjerio.

Uz drugarski pozdrav
Horvat Božidar

Tražimo — nalazimo

Robert Furatenberg Cara Lazara 74 I/7 24000 Subotica ima računar C-64 sa kasetolonom i potreban mu je program koji „izračunava položaje planeta sideralno vreme i astrološke kuće" kako navodi u svom pismu Mada opis nije baš najjasniji, pretpostavljam da se radi o programu koji treba da zameni tablice alamanda Ako neko ima nešto ovakvo neka se javi

Vlado Badanjak, Ivana Gorana Kovačića 69 55400 Nova Gradiška ima drugih probleme Poseduje 54 lastera tip T4 koji su mu „višak" a „manjak" su mu memorijski čipovi i procesor HD64180 (pretpostavljamo da je sve to vezano za gradnju tim-a) Ukoliko ima te „kompatibilan" manjak i višak javite mu se

maćeg autora jedne od verzija legendarnog pakmana U folderu IGRE su tri male igrice s tim što je program NJIHALO na granici igra i obrazovnog softvera Ovaj program daje simulaciju matematičkog klatna i uz pogodno odabrana parametra možete dobiti pravo fraktalno kretanja Treći folder nosi ime TETRIS i sadrži istoimenu igru napisanu za GFA BASIC Dat je i BAS i LST fajl sa kojima možete raditi sve što poželite

Javni softver

Programe za atari ST opisane u ovom broju u tekstu o javnom softveru dobijate u fajlu RAČUNARI 52 Programi iz ranijih brojeva su takođe raspoloživi Programe za PC „Računari" ne distribuiraju — za njih se možete obratiti Aleksandru Jovanoviću Mokranjčeva 30 18000 Niš koji drži VIK BBS (018 44 873)

Do sada objavljeno

Broj fajlova se PD softverom je dostigao broj u kome je praktično nemoguće vršiti snimanja fajl po fajl u razumnom vremenu Zbog toga se već mesec dana vrši **snimanje samo kompletnih diskova** Diskovi će biti formata 83 puta 10 Svaki disk sadrži i program **AR** za arhiviranje i dearhiviranje

disk

RAČUNARI 001
RAČUNARI 45
RAČUNARI 46
RAČUNARI 47
RAČUNARI 48
RAČUNARI 002
RAČUNARI 49
RAČUNARI 50
RAČUNARI 52
ST_NEWS 001
ST NEWS 4_2

ST NEWS

je PD magazin koji se izdaje u Holandiji na engleskom jeziku Izlazi 6 do 7 puta godišnje ovisno o količini teksta koji stigne redakciji Čitav magazin je organizovan kao jedan program iz koga se na ekran pozivaju tekstovi (koji se mogu i izdvojiti i spremiti disk ili odštampati) uključuje i isključuje muzika itd Kompletan sadržaj magazina obično zauzima 400 do 500 K tj oko 250 K u komprimovanom obliku Za ST NEWS uvek treba poslati poseban disk neovisno o diskovima koja šaljete za PD softver

Postupak slanja diska

Disk pošaljite u običnoj maloj koverti a uz njega i još jednu malu kovertu sa vašom adresom i zalepljenim markama dovoljnim za povratak diska (službenica na pošti će vam najbolje reći koliko maraka je potrebno) Ukoliko želite da vam se disk vrati **preporučeno** (što je mnogo sigurnije) morate i za to nalepiti odgovarajući iznos poštanskih maraka

Ukoliko posedujete javni softver na svojim disketama bilo bi lepo da ga podelite sa ostalim čitaocima jer tako će svi imati više programa Jednostavno nam namojte poslati praznu disketu već na nju stavite javni softver koji imate Javni programi **domaćih autora** su posebno dobrodošli

Šaljite **isključivo KVALITETNE DISKETE** U suprotnom vam se može desiti da umesto programa dobijete disketu sa nalepnicom „invalid medium" Ovo se posebno odnosi na diskete koje se tresu i klopoću kao voz na šinama

Raspakivanja programa

Na disku imate program AR koji služi za raspakivanje i pakovanje Pretpostavimo da imate otvoren RAM disk D Na njega prekopirajte fajl RAČUNARI.nn gde je nn broj Potom na nekom disku u A stvorite folder RAČUNARI nn Pazite da na disku imate dovoljno prostora (stepen kompresije je od 45 do 55 procenata) Potom pokrenite AR ubacite u A disk sa folderom RAČUNARI nn pa zadajte

from **D \RAČUNARI nn**
to **A \RAČUNARI nn * ***

Potom selektujte opciju **unpack** i pritisnite RETURN Program će ostatak obaviti za vas Ukoliko imate dovoljno velik RAM-disk možete stvar obrnuti pa raspakivati direktno sa diska koji ste dobili u folder RAČUNARI nn na RAM-disku

MALI OGLASI

Običan mali oglas do 10 reči staje 80 000 dinara a svaka dalja reč 8 000 dinara. Centimentar uokvirenog oglasa staje 100 000 dinara s tim što se mogu zakupiti najmanje 2 cm i što oglas ne može biti veći od 5 cm. Za uokvirene oglase preko 5 cm važe komercijalne cene.

Postupak je jednostavan: pošaljite redakciji tekst oglasa zajedno sa foto-kopijom uplatnice kojom ste na žiro račun RO BIGZ 60802-603-23264 (za Male oglase u „Računarima") uplatili odgovarajući iznos.

Prodajem Commodore-64* sa originalnim kasetofonom i kolor monitorom — 700 DEM. Odgovaram samo na pisane ponude 21241 Kać Moše Pijade 20 Grujić

JM HARDWARE za CPC 464 664 6128 senzorski joystick EPROM BOX proširenja ispravljači kabeli light pen. Sve informacije na tel (058) 666 526 ili 585 368

Kompletna oprema za stono izdavaštvo računari programi uputstvo ju-slova. Podrška kod uvođenja. Tel (011) 454-074 od 16—18h

---

Veoma povoljno prodajem „SCHNEIDER CPC-464" i disk DDI-1 Tel 054/842 536

COMPUTER SERVICE
VIII Vrbik 33a/II 41000 Zagreb tel 041/535 277
od 10 do 12 i od 15 do 17 sati
— SPECTRUM COMMODORE ATARI AMSTRAD PC XT PC AT
— brzi i kvalitetni popravci
— memorijska proširenja
prodaja kompjutera monitora flopy drive
— kablovi za povezivanje sa printerom monitorom televizorom i drugom periferijom
— ZX centronic interface joystick interface
— C64 eprom moduli centronic kabel
— rezervni dijelovi za kompjutere
Tražite besplatni katalog dodataka za kompjutere i rezervnih dijelova

---

Za IBM PC nudimo saradnju na području hardvera i softvera pojedincima i radnim organizacijama. Zatražite besplatni katalog na disketi

Tel (054)/885 104

ATARI ST HARDWARE
— HARD DISKOVI I MONITORI
— SM 124 SA TRI REZOLUCIJE DISKETNE JEDINICE 3.5" 12 <3.5
— DISK 5.25 ZA PC DITTO
— SF 354 KAO DVOSTRANI DISK VIDEO-DIGITALIZATORI EPROM PROGRAMATOR SOUND SAMPLERI
— PROŠIRENJE MEMORIJE SCANNER PRINTER I KOPIR U JEDNOM
— MODEMI
TOS — NAJNOVIJI (1.4)
— SCART I KOMPOZITNI KABEL
— I JOŠ MNOGO DRUGOG
BESPLATAN KATALOG
ŠKROBAR P.P. 39 42300 ČAKOVEC 021/394 042
042/817 596 041/511 139

Prodajem za IBM PC Hard disk Segate ST 412 floppy disk 1.2 Mb grafičku karticu Herkules I/O karticu sa dva paralelna i dva serijska porta

Tel (054)/885 104



HARDVERSKA ZAŠTITA PROGRAMA OD NEAUTORIZOVANOG KORIŠĆENJA

-priključuje se na paralelni izlaz PC-a
-transparentna za štampače
-interfejs rutine za MS PASCAL I FORTRAN TURBO PASCAL I CLIPPER
-jednostavno uključivanje u programe
-različiti sklopovi za autore

Vesna Marković Kolarova 14
Zagreb (tel 041-227-381)

DISKETE 5.25"

DS/DD (360 Kb za XT)
DS/HD (1.2 Mb za AT)

071/214 319 111 626-519

Prodajem vrlo povoljno sljedeće računare i štampače
— VIKTOR AT (286) 1 flopi 1.2 Mb HD 30 Mb 640KB RAM PEGA kartica Amber monitor fol 6000 DEM
— EPSON LQ 850 2000 DEM
— SCHNEIDER PC1512 1 flopi 640kb HD 21 Mb Hercules i CGA kartica Mouse dodatni ventilator + štampač STAR SG 10(NLQ) 3500 DEM
— SCHNEIDER PC1512 1 flopi 640kb,HD 21 Mb CGA kartica + štampač STAR NL 10 (NLQ) 3500 DEM

Svi računari i štampači su sa carinskom deklaracijom. Vrlo povoljno za RADNE ORGANIZACIJE uz zakonski način plaćanja. Zvati svaki dan od 16—20h. Savić Goran M Tita 151 — 22/4 75000 Tuzla tel (075)223 866

Povoljno prodajemo računarski sistem PDP11/40. Informacije tlf (085)22-913

„UNISOFT"
Servisiranje sklapanje i prodaja kompjuterskih sistema PC /XT /AT /386 /486
Ugrađujemo HARD diskove FLOPY diskove grafičke kartice proširenje memorije modema spajanje u mreže i ostalo
Pružamo savjete u vezi izbora računara cijene komponenti prije konačne odluke obratite nam se sa potpunim povjerenjem
Izrada programa po želji za radne organizacije i privatnike
Nudimo potpuni INŽINJERING za korištenje računala sa garancijom
„UNISOFT" servisiranje kompjutera i izrada software-a Baksa Zdenko I Milutinovića 34 41040 ZAGREB tel 041/254 581

Želite li znati nešto više o Vašem IBM PC-u ili kompatibilcu, a ne samo MS DOS i Basic naručite knjigu

Povezivanje na IBM PC

Priručnik bez kojega se nemože

Interna arhitektura računala, Karakteristike sistemskog BUS-a, Interupt-i (prekidi)-kontrola i korištenje, D M A mapa ulazno-izlaznih adresa, vanjsko programiranje interupta i timing-a, povezivanje sa specijalnim karticama i uredjajima, kako programirati ulaz-izlaz, hardware i software za testiranje
400 stranica, format 17x23 cm
Cijena 95 000 din
Knjigu možete naručiti direktno od izdavača dopisnicom ili na tel. 055-24 11 35
Plaćanje pouzećem.
Elektronika BARBARIĆ
Slav.Brod Augusta Cesarca 15a

---

računari

su uvek aktuelni

Ono sto nema u novom, potražite u starim brojevima

IBM-SPECTRUM
programi za
LOTO
011/600-116

CADET
CLIPPER APPLICATION DEVELOPMENT TOOLS

Alat za razvoj aplikacija u dBase Clipperu generator potprogramska biblioteka, unos podataka uslužni programi tabeliranje
Povećanje produktivnosti 5 — 10 puta
Značajno unapređenje kvalitete ergonomije korisnika i fleksibilnosti aplikacije
Cijena 4.0—17.3 M dinara (opcija plaćenja 7 1989.) zavisno o kompletnosti i vrsti korisnika
Informacije Autorska agencija 41001 Zagreb Preradovićeva 15 pp 379 tel (041)445 428

Razbarušeni sprajtovi



Uređuje Slobodan Perović

Rubrika se realizuje u saradnji sa redakcijom emisije „Čip i sedam jarića", koja se svake subote, emituje na Prvom programu Radio Beograda
Zajednički telefon.
011/339-070

Gotovo je gotovo

OCEAN RANGER

Još jedna u nizu simulacija iz drugog svjetskog rata. Na početku nakon pozdravne-upozoravajuće poruke (koja se mijenja pri svakom učitavanju) treba izabrati jedan od ponuđenih zadataka („level") ito osnovne borbene dužnosti redovne službe, rad u CIA nacionalne bezbjednosti te mjesto gdje će se odigravati vaša misija („theater") Perzijski zaliv, jugoistočna Azija, centralna Amerika i Beringovo more

Na kraju („briefing") odlazite po dalje instrukcije koje pregledate sa „next" i „previous" i mijenjate sa „abort" a u prethodni meni vraćate sa „admin". Možete odpuniti brod oružjem i gorivom pri čemu treba da pazite na ciljeve koje treba da uništite i njihovu udaljenost od baze te shodno tome povećate ili smanjite količinu pojedinog oružja, odnosno goriva. Pritiskom na „ship" naazad možete krenuti u svoju misiju

U toku igre mnoge poruke pojavljivaće se u redu iznad, dajući vam često korisna obavještenja. Za određivanje kursa (idući u smjeru kazaljke na satu) imate

0 — sjever 90 — istok 180 — jug 270 — zapad odnosno

0 — 90 sj. istok 90 — 180 j. istok 180 — 270 j. zapad 270 — 360 sj. zapad. U sredini se nalaze dva monitora na kojima možete imati 6 različitih prikaza i to: uvećanu mapu bližeg područja, sliku objekta koji vas napada, prozor na more, stanje oružja, pogođene ciljeve, koordinate misije (PR primarna misija, SC-sekundarna misija, H-polazna baza) i radar. Najbolja kombinacija je da na jednom monitoru imate cilj ili mapu, a na drugom radar. Ostalo dobijate sa CRSR tipkama

U slučaju napada izaberite odgovarajuće oružje i kada nišan „pobijeli" sa fire ispaljujete projektile

Za mine se može koristiti samo 76 mm top. One su njedno i najopasnije. Brodove i avione koji vam se približe suviše blizu može te uništiti samo sa topom

Oštećenja popravljate prebacujući ljude na mjesto oštećenja koji su ovdje predstavljeni u vidu malih figura. Najopasnija su oštećenja magacina ili trupa broda („hull") i mašina („engine") i ako pređu brojku od 200 ubrzo tonete. Oštećenja kompjutera i oružja u težem slučaju mogu onemogućiti gađanje

Evo i pregleda funkcija ostalih tipki

A — acceleration — ubrzanje, pomaže da brže stignete do cilja. U normalno stanje se vraćate ponovnim pritiskom na „A" ođk nailaskom na neprijatelja kompjuter sam smanjuje brzinu i vraća vas na komandni most

INST/DEL — mapa sa ucrtanim koordinatama

CLR/HOME — mapa bez koordinata

— kompjuter sa mapom

— — kompjuter sa pogledom na pučinu

+ — prikaz broda sa oštećenjima

0 — prikaz oštećenja i njihov popravak

8 — dizanje i spuštanje sidra

6 — ECM sistem za navođenje

5 — mamac za neprijateljske vođene projektile (chaff) (o tome vas obavještava kompjuter)

4 — dubinske bombe (depth charge) — za podmornice

3 — harpun (harpoon missile) — za brodove

2 — rakete more-vazduh „morski vrabac" (sea sparrow) — za avione

1 — 76 mm top (76 mm canon)

← — vraća vas na most

F — rezerve goriva

S — pauza

F1 — prednji pogled

F3 — lijevi pogled

F5 — desni pogled

F7 — zadnji pogled

Ukoliko budete potopljeni ili ostanete bez goriva u savezničkim vodama postoji šansa da vas spasu. U suprotnom postajete ratni zarobljenik. U bazu se možete vratiti u bilo kojem trenutku da popravite brod i dopunite gorivo i oružje ili pak da prekinete misiju i pokušate našto drugo

Nakon završene misije slijedi izvještaj i ukoliko se budete dobro pokazali slijedi i počasne trake (značka) ili odlikovanje. Takođe zavisno od ukupnog skora povećaće vam se i čin

Prije početka svake nove misije vaši rezultati i uspjesi će biti zapisani na disk, tako da kasnije možete nastaviti igru i dalje graditi karijeru

I za kraj, na drugoj strani diskete (game disk) prvi fajl vam omogućava da napravite backup game diska, te da u slučaju da „poginete" možete snimiti taj status dok na „originalu" imate i dalje stari. Mnogo sreće!

Amer Užičanin, Tešanj

Razbarušeni sprajtovi

SWORDS OF BANE

Na početku igre izaberete nivo težine, to jest broj zlatnika. Što je nivo veći, zlatnika je manje. Kada izaberete nivo, počinjete sa kupovinom vojske. Možete kupiti čarobnjake sa i bez štapa, strelce, kopljanike sa i bez štita, mačevaoce sa i bez štita i obične vojnike, takođe sa i bez štita. Čarobnjaci i strijelci mogu gađati na određenu udaljenost, ali su slabi u direktnim okršajima, dok ostala vojska ne može gađati, ali je odlična u direktnim okršajima

Kada kupite vojsku, počinje borba. Borite se protiv raznoraznih čudovišta. Cilj igre je ubiti glavnog demona. Njega ćete prepoznati lako što se u početku ne pomjera. Predlažem vam dva načina igre za pobjedu i to

1. **Obrambeni način.** Uvedite svoju vojsku u malu kuću tako da su čarobnjaci u sredini kuće postavljeni tako da mogu gađati na sva tri ulaza, a ostalu vojsku postavite pored ulaza s unutrašnje strane. Pošto su vrata uska čudovišta će moći ulaziti samo pojedinačno. Zbog vojske neće moći proći u kuću, pa je to prilika za čarobnjake

2. **Napadački način.** Krenite u frontalni napad tako da glavninu svojih snaga pošaljete direktno na neprijatelja, a da se 2—3 čarobnjaka zaobilaze neprijatelje i ubijete glavnog demona.

Vedran Senčić, Vinkovci

Usijani džojstik

Za one koji su izgubili svaku nadu, evo informacije koja će ih možda i prijatno iznenaditi

Po poslednjim informacijama koje smo dobili iz FGP RTB igra „Vertigo" u verziji za „Sinclair" pojavila se u prodavnicama

Vlasnici C 64 moraće i dalje da se strpe

Što se tiče „starija" i „amige", redakcija emisije „Čip i sedam jarića" stupila je u kontakt sa firmom „Acivision" pa ima izgleda da se i tu nešto desi

* * *

Dobili smo oštar protest Branislava Jovanovića iz Bora, jer je izostao potpis ispod njegovog objavljenog priloga za koji takođe nije dobio honorar za prikaz igre „Murder on the Mississipi" („Računari 48")

Dabome da se izvinjavamo i da ćemo stvari dovesti u red

* * *

Poduže pismo poslao nam je Leon Marenković iz Siska. Kao prvo traži pomoć za igre „Frankenstein's bride", „Big trouble in little China", „Sorcerer lord" i „Gothic" (za ovu poslednju detaljan opis je objavljen u jednom od ranijih brojeva „Računara")

Pokoni šalje i nekoliko cikа

Ukoliko u igri „Netherworld" pritisnete taster „Q" prelazite u sledeći nivo, a u igri „Mega adicta" istu funkciju ima taster F3. Takođe prilaže i šifru 210319 19 koja u igri „Detective" otvara sef koji je u sobi sa kaminom skriven iza slike

I na kraju žali nam se što smo ga — omaškom — potpisali sa Ledon umesto Leon

* * *

Polemicu za pouk za igru „Batman" (C—64) raspisao je Igor Isaković iz Novog Sada. Nadamo se da će se neko odazvati

* * *

„Šta vam bi da objavite prikaz igre Platoon u kojoj Vijetnamci stradaju kao muhe?" piše nas Gordan Krstičević iz Metkovića.

Gordane, potpuno si u pravu — za ovaj propust, koji nije u skladu sa našim kriterijumima, izvinjavamo se svim čitaocima „Računara" istovremeno na našu omašku stavljamo i igru „Hostage", čiji smo prikaz dobili nepotpisan

* * *

Na kraju podsećamo vas na zajednički telefon „Računara" i emisije „Čip i sedam jarića" — 011/339-070 — pa ako nešto nije jasno prilika je da se javite

TRUCK
(„atari ST")

Igra se sastoji iz tri dela u kojima vozite kamion, ali sa različitim ciljem

GAME 1 je klasična trka. Trkate se sa još 11 kamiona i jedini cilj vam je da stignete prvi. Vaš kamion ima 2 brzine i može da dostigne brzinu i od 220 km/h. Voze se 4 kruga, a ograničenog vremena nema. Pri većim brzinama kamiona odbija poslušnost, a imate samo jedan život

GAME 2. Ovde je cilj da što duže vozite na dva točka. U početku treba voziti brzinom od oko 90 km/h, ali čim se „prevrnete" na dva točka brzinu treba povećati od maksimuma. Ravnoteža se održava kratkim trzajevima džojstika.

GAME 3. Vozite između čunjeva i cilj je da za što kraće vreme pređete ceo poligon. Čunjevi se mogu rušiti, ali se poeni dobijaju samo za nesrušene čunjeve

Jedini pravi biser u igri je muzika koja zaslužuje najveću ocenu

Nikola Markićević, Paraćin

Mala istorija igara (1)

Igrali se sprajtovi

Gotovo svako od nas je bar jednom odigrao igru na nekom od automata za zabavu. Ove opasne naprava veoma su zarazne i u slučaju da ste dugo u kontaktu s njima, dolazite u opasnost da budete inficirani jednom veoma opakom bolešću — besomučnim trošenjem novca. Vremenom novac ćete trošiti sve manje, a igrati i zabavljati se sve više. U ovoj maloj seriji članaka pokušaćemo da obuhvatimo čitavo tržište automata za igru — od legendi, pa do najnovijih ostvarenja.

Naravno, svi smo čuli za igre PAC MAN, INVADERS koje su donele milione dolara i zadavljale milione ljudi. Međutim ima još igara u toj prvoj fazi ere automata. Jedna od njih je CENTIPEDE u kojoj je cilj bio da uništite gusenicu koja se spušta sa vrha nadole. Trebalo je odbiti se i protiv paukova koji su besomučno napadali. Uprkos uspehu ovog automata on nikada nije doživeo svoj drugi nastavak.

Stare legende

DEFENDER je na tržište automata izbacila firma „Atari". To je klasična pucačka igra kojih sada ima na hiljade, ali je u ono vreme predstavljala pravi bum. Posebno je oduševljavao radar koji je pokazivao položaj broda i neprijatelja a koji danas ima svaki drugorazredni program. Može se reći da je njegov sledbenik SCRAMBLE koji je za osnovu imao sličnu ideju i izvođenje. U to vreme je i firma „Nintendo" napravila još jednu legendu DONKEY KONG. U toj igri ste u ulozi radnika Marija kome je majmun Kong oteo devojku i polako se penje na jednu zgradu. Mariju pomaže i čekić što određeno vreme uništava buriće koja Kong baca. Nivoa je u originalu bilo četiri, a posle četvrtog sve počinje od početka. Ova igra je imala više nastavaka na automatu KONG JR, KONG SR ali nijedna nije doživela uspeh svoga prethodnika.

Na temeljima Donkey Konga „Nintendo" pravi jednu sličnu igru koja takođe doživljava uspeh MARIO BROS. Nju i „Atari" prodaje na svom kućnom automatu za igru Atari 26000 ali pošto je kupio licencu od „Nintenda". Posle nekog vremena ljude je ponovo zarazila jedna svemirska igra — MOON CRESTA koja je prerađivana i za kućne računare, a prva ima za ideju nadograđivanja osnovnog broda i dodavanje jačeg oružja. Kada smo već kod svemira nikako ne možemo a da ne pomenemo mega hit PHOENIX koji je i kod nas bio veoma popularan tako da na njega nećemo trošiti previše reči. Toliko o starim legendarnim izdanjima koji su od industrija automata napravila milionsko multimedijsko tržište. Oni su, kako se to kaže, izgradili temelje za sve novije i što je najvažnije sve savršenija automate.

Prvo je programerska kuća „Elite" došla od zaključka da će se hitovi sa automata odlično prodavati i na tržištu kućnih kompjutera pa na tržište izbacuje nekoliko hitova koje ne smemo preskočiti.





Prvi je COMMANDO koji vas vodi u ulogu ratnika što se probija kroz neprijateljski logor od „Tecma" je otkupljena licenca za njihov hit BOMB JACK koji je postigao veliki uspeh. Na kraju najbolji je bio GHOSTS AND GOBLINS koji je stvorio „Japan Capsule Computers". Ova igra vodi vas u davna vremena vitezova ali tema je ista kao i u Donkey Kongu. Neko vam je oteo devojku a vi je morate spašavati. Treba napomenuti da ova igra na automatu izgleda zaista izvanredno.

Svemirske teme

Posle ovih srednjovekovnih tema selimo se u krutu sadašnjost sa igrom GREEN BERETS koju zbog njenog sadržaja ne podržavamo ali je moramo pomenuti. Stara izlizana tema pojavljuja se još jednom uz odličnu realizaciju u igri firme „Sega" WONDER BOY. Cilj vam je da u ulozi čudesnog dečaka oslobodite svoju bavojku koje je zatočena u džungli. Njenu konverziju napravio je „Activision" i na „spektrumu" imala je veoma dobru grafiku ali se svaki nivo morao učitavati posebno što je dovodilo do velikog gubljenja vremena.

Da igra sa automata ne more da uspe na kućnim kompjuterima pokazala je igra DRAGON'S LAIR koju je konvertovao „Software Projects" sa legendarnim Metju Smitom (Mathewom Smithom) na čelu to joj nije pomoglo da izbegne najoštrije kritike i ocene. Setimo se i QUARTETA koga su mogla igrati dva igrača na automatu. Naravno proizvođač je „Sega". Kad smo kod igara za više igrača setićemo se i GAUNTLET-a koga je na kućne kompjutere izvrsno prebacio „U.S. Gold". Kasnije ova igra proglašena je za igru godine a usledio je i nastavak — GAUNTLET II. Međutim svemirske teme su oduvek bile u modi što su potvrdile igre SLAP FIGHT, MAG MAX i TERRA CRESTA. Sva tri igre je za kućna kompjutera preradila britanska firma „Imagine". Treba pomenuti i jednu od retkih 3D igara CRYSTAL CASTLES. Zadatak u igri bio je da pokupite sva dijamanta na određenom nivou ledenog zamka. Posle toga prelazili biste na sledeći nivo i tako u beskraj.

„Elite" ponovo stupa na scenu sa prepravkom izvrsnog SPACE HARRIER-a koga je stvorila firma „Sega". Leteći sa mlaznim rancem treba da se odrite protiv ogromnih zmajeva i glava i izbegavate kamenja koja se nalazi na putu. Igra na „spektrumu" odlikovala se izvanrednom grafikom i svetlosnom brzinom.

Negde u vreme posle izdavanja Space Harrier-a ljudi iz firme „Taito" dolaze na genijalnu ideju da legendarnu igru odbijanja loptice od zida preprave. Ovoga puta tehnički savršenija i modernija. Dodali su mnogo novina, neprijatelja, letećih tanjira i nivoa. I tako je nastao legendarni ARKANOID koji je zbog velikog uspeha doživeo nastavak — ARKANOID II (Revenge of Doh). Najpametnija je bila opet firma „Imagine" koja je otkupila licencu i briljantno konvertovala obe igre na kućne računare.

Posle Arkanoida veću pažnju izazvalo je pojavljivanje igre NEMESIS koja je u to vreme bila tehnički perfektno urađena i koja danas služi za primer ostalim igrama koje se prave na sličan način. Ova igra je zbog svog velikog uspeha doživela još nastavaka o kojima će biti reči kasnije. Da ne zaboravimo proizvođač je „Konami". Sledeći je RYGAR koji je na kućne računare prebačen na lako sjajno. A posle njega dolazi prava poslastica i za ljubitelje automata i za ljubitelje kućnih računara — KARNOV. Original je napravio „Data East" a konverziju „Activision". Nalazite se u ulozi abnormalno jakog Rusa koji se bori sa svakakvim demonima ne bi li došao do mape koja vodi do skrivenog blaga. Sledeći je RASTAN koji vas vodi u ulogu kralja ratnika Rastana. On se kao i Karnov mora probiti kroz pregršt nivoa da bi dosegao kraj igre što je nekolicini i pošlo za rukom. Da ne zaboravimo original je „Taitov" a licencu je otkupio „Imagine".

Utabani tragovi

Ugledavši se na „Technosov" DOUBLE DRAGON igru čiju je konverziju za osmobitne računare napravio „Melbourne House" poznata japanska firma „Taito" pravi trodimenzionalnu tučnjavu iz ptičje perspektive pod nazivom KAGEKI. Igra se odlikuje odličnom grafikom i krupnim veoma dobro nacrtanim i animiranim likovima. Radnja se odvija u Hong-Kongu i vaš zadatak je da savladate celu bandu jednog po jednog. Naravno dok se vi tučete sa jednim ostali članovi bande navijaju u pozadini ekrana. Pri tom je najglasniji vođa sa kojim ćete imati najviše muka na kraju — naravno ako od njega dođete jer je igra veoma teška.

Posle ogromnog uspeha Nemesis se izlazi nekoliko njegovih nastavaka od koga je najuspeliji VULCAN VENTURE. Igra je pravo savršenstvo i grafika i animacija su našlo od tada neviđeno na automatu. U principu misija vam je ista kao i u ostalim delovima s tim što možete da nadograđujete brod još novijim i naprednijim oružjima. Takođe su dodali i novi neprijatelji koji su po starom običaju veličina celog ekrana — planete, plice, ogromne glave, svemirski brodovi itd. Igra veoma podseća na fenomenalni R—TYPE koji je ušao u legendu posle izvrsnog prebacivanja na kućne računare posebno na „spektrum" gde je postavio nova standarda za igre ovakvog tipa. Firma koja je izbacila R type je japanski „Irem Corporation".

Ideja je klasična, treba se odriti protiv neprijatelja u svemiru i na kraju svakog nivoa uništiti ogromnog neprijatelja čuvare tog nivoa.



I pored neverovatno dobre grafike i veoma brze i dinamične i ove igre bi se ubrajale u tog ranga da nama neke stvarno originalne probleme za rešavanje. Da ne zaobilazimo izvrsnu konverziju na „spektrum" izvršili su programeri Bob Pejp (Bob Pape) grafiku je uradila kuća „Mak CG" a igru je izdao „Electric Dreams". Treba pomenuti i koje su igre sledile utabane tragove R types i Nemesisa SALAMANDER koga je izdala ista kuća kao i Nemesis i SIDE ARMS koga je konvertovala firma „Capcom". U suštini igra je bila odbro urađena i u tadašnje vreme iskakala je iz proseka.

Usijani drumovi

Posle namašlovitih borbi protiv sovjetskih i korejskih jedinica „Namco" je rešio da obrne list pa vas u igru ROLLING THUNDER stavlja u ulogu specijalnog policajca koji se odri protiv zloglasne rasističke organizacije Kju-Kluks-Klan. Igra je dinamična i dobro urađena i u svoje vreme bila je veliki hit. Druga igra sa sličnim izvođenjem ali drugačijom tematikom (radnja se odvija u svemiru) je „Teitov" SOLDIER OF LIGHT (u Japanu gde je ova igra i nastala zove se XAIN D SLEENA). Ideja stara i dobro znana: vaš zadatak je da istrebite sve ljudine sa četiri planete. Treba li pominjati da je vaš zadatak sa onim brojem života koje je predvideo autor gotovo nemoguć. Na temu Gauntleta „Data East" je izbacio na tržište SCHAKLED koji je prebačen na osmobitne mašine i prošao gotovo nezapaženo. Međutim koplja su se lomila oko „Capcomovog" STREET FIGHTERA. Neki su govorili da je neverovatno dobro urađen sa odličnom grafikom i ogromnim sprajtovima a drugi su program nipodaštavali govoreći da je neverovatno lak i da se na igre više od nekoliko sati. Šta je od svega toga istina — ostavljamo da prosudite sami.

Sećamo se prve laste „Atarijevog" POLE POSITION-a koji je u svoje vreme imao izvanrednu brzinu i veran trodimenzionalan prikaz bolida. Posle ovog programa bilo je nekih neuspelih imitacija vožnje ali su gotovo sve prošle nezapaženo. Dugo se čekalo na pravog naslednika a onda je odšao ENDURO RACER koji je konvertovan i na kućne računare pa je napravio pravi bum. Radi se o trci motorima po neravnom putu punom prepreka osim kojih su vam smetali i ostali vozači. Posle ovog programa uvedeni su novi standardi za igre ovakvog tipa. Kada se pomislilo da će se dugo čekati da neka nova igra pređe te standarde pojavio se izvrsni SUPER HANG-ON koji na automatu izgleda zaista zadivljujuće. Na originalnom automatu nalazi se konstrukcija motora i pravi upravljač kojim se kontroliše motor na ekranu. Na tom automatu u Grčkoj bio je po pravilu najveći red a nakolicini bi se oteo i vrisak posle eksplozije koja je i te kako realistična. Konverzija na kućne računare je odlično urađena a najbolje od svega urađeno je penjanje i spuštanje na brdo.

Beskrajni nivoi

„Atari" posle ovoga izbacuje ROAD BLASTER gde pored sumanute vožnje morate pucati na neprijatelje oko sebe pa čak i na helikoptere koji vam dolaze iza leđa. Sve je to pomalo podsećalo na stari dobri SPY HUNTER ali to nekima nije smetalo da postave fantastične rekorde sa ovom igrom.

Greh bi bio krenuti dalje, a ne pomenuti legendarni OUT RUN koji je kao i Super hang-on proizvela „Sega". Out run je najprodavanija igra svih vremena na kućnim kompjuterima (otišla je u 260 000 komada i još se prodaje) a i na automatu je veoma rado igrana. Cilj vam je da svoju plavokosu devojku provozate svojim ferarijem (nije Testarossa) kroz različite predele za razumno određeno vreme. Šta na kraju dobijate od nje kao nagradu drugi put. I da pomenemo najnoviju igru koju već uveliko reklamira „Ocean" — WEC LE MANS koja vas stavlja u ulogu vozača bolida. Šuška se da će igra po realizaciji prevazići i Hang On.

Za kućne kompjutere uveliko se najavljuje „Tecmov" GEMINI WING igra na bazi 1942: ekran se skroluje nadole a vaš zadatak je da uništite sve što se kreće. Samo za zagrižene ljubitelje pucnjave! Međutim „Namco" se odlučio za rizičan potez i obnovio staru ideju iz igre GALAXIANS uz odličnu grafiku i realizaciju i stvorio igru GALAGA 88. Ukoliko ste voleli Galaxiane ovo je prava igra za vas jer su nivoi produženi do u beskraj. SNK je prešao na sportske simulacije dodajući im ispred pravog imena pridev FIGHTING. Tako nastaje FIGHTING GOLF. Međutim ovde nema nikakve odlike ovo je obična simulacija golfa kojih smo imali kao pleve na našim računarima.

Do sledećeg nastavka („Na štavičemo se" rekao bi pokojni Ljuba Moljac) džeparac u ruke — pa do prvog automata.

Vladimir Janković

Poukova mreža

1 death wake poke 3 poke 29292 0
2 jack the nipper 2 poke 36670 173
3 ghostbusters name andy credit 777
4 hercules poke 3805 169 poke 3807 0 poke 3807 234 sys 2304
5 miner 2049 poke 9450 173 poke 2652 165 sys 2471
6 loco poke 26901,6
7 arcanoid poke 4482 44 poke 8102 44
8 choplifter poke 8011,173
9 kung fu master lives poke 34142 x.poke 34184 x poke 34208 100.sys 32768
10 zeppelin poke 18546 44
11 international karate
poke 7154 x poke 2820 x poke 2831 x for a=3917 to 3919 poke a 234 next for a=3923 to 3923 poke a 234 next
12 bmx simulator poke 13917 0 sys 4096
13 druid poke 15324 234 sys 4096
14 asterix poke 8774,173
15 venom strikes back kodovi petals of doom mayhem trnsmogrify
16 yogi bear oke 6478,0.sys 2064
17 pink panther poke 15324,225 sys 3825
18 ollie and lissa poke 11238 180
19 bouder dash construction kit staza koju napravite startuje se opcijom „start game"
20 guertel poke 42030 173 poke 1720 173 poke 81232,173:poke 8209 0
21 dig dug poke 10465 234 sys 2071
22 hard hat mak poke 16687 173.sys 2063

Nikola Jovičić Aleksinac

BLAST BALL (C-64)

Nakon velikog uspjeha igara kao što su Arkanoid, Impact, Batty stigla je još jedna istog tipa. Nakon što smo se gušili u raznim pucačinama slično se događa sa igrama ovog tipa. Gotovo svakog mjeseca napravi se barem jedna ovakva igra a rezultat je poznat. Kvalitet igara je sve lošiji. Ipak možemo reći da svaka igra donosi nešto originalno. Tako je i sa ovom. Sigurno vam je poznato da je cilj ove igre porušiti što više ciglica pomoću strašne loptice i tako preći na sljedeći nivo. Igra podsjeća na „Arkanoid 4" u verziji za CPC 464. Kao i u svim sličnim igrama i u ovoj vam se s vrha spuštaju razne poboljšanja označena slovima. Poboljšanja vam pomažu da što prije završite nivo. Njih nije obavezno skupljati a pomažu vam na slijedeće načine:

E — proširuje se reket kojim odbijate lopticu

T — Omogućava vam da igrate s tri loptice odjednom pa tako možete brže završiti nivo

C — loptica se ne odbija od reketa ako vi to ne želite. Tako možete odlučiti koju ćete ciglicu uništiti

S — usporava čitavu igru

L — poništava sva poboljšanja

Novost u ovoj igri je mogućnost konstruisanja nekog od nivoa. To vam omogućava EDITOR MOD a u njega ulazite pritiskom Control i E tastera. Control i R tasteri zajedno pritisnuti resetiraju igru a Control i a tasteri vraćaju vas u glavni meni. Grafika i zvuk u igri su zadovoljavajući a i ideja je dosta dobra. Autori ovog teksta (posebno prvi potpisani) su veliki obožavatelji igara ovog tipa i igru preporučujemo svima.

Saša Kusanić, Igor Gomezalj, Slavonski Brod

SUPER SPORTS – THE OLIMPIC CHALLENGE

Još u prvom delu ove izvanredne igre saznajemo da će nas komentator Gilbert voditi kroz sve discipline SUPER SPORTS-a. Prvo treba odrediti broj i imena igrača. Može se takmičiti najviše 4 igrača. Prvi je sudeći po izgledu Englez drugi Nemac treći Amerikanac a četvrti verovatno Francuz.

Discipline su:

CRACK SHOT — streljaštvo. Za 60 sekundi treba pogoditi sve što se miče i ne miče na ulici ali ne i mačke.

DARE DEVIL DIVE — skokovi u vodu. Svaki takmičar ima 3 pokušaja za pravilan skok. Računa se onaj koji donese najviše bodova. Najmanja visina je 40 a najviša 400 metara. Ocenjuju se visina i stil. Kombinacijom tastera može se napraviti čak devet figura u vazduhu.

SLATE SMASH — u ovoj disciplini cilj je porazbijati što više cigli koje drže dva sumo-rvača. Poeni se dobijaju i za neobične udarce.

CROSS BOW — na žalost po učitavanju ove discipline računar blokira.

WATER ASSAULT — u ovoj disciplini cilj je stići do kraja trke koja se odvija ispod morske površine. Ekstra poeni se dobijaju za skupljanje novčića i provlačenje kroz automobilske gume. Da dođete od neophodnog svežeg vazduha ometaju Vas hobotnice ribe i buzdovani.

Slobodan Bjelica Novi Sad

STAR TREK

(„C—64")

Posle izlaska velikog broja igara sa filmskim temama (ROGER RABBIT SUPERMAN RAMBO III) uzimajući n obzir veliki uspeh filma STAR TREK možda je bilo logično očekivati istoimenu igru Njena ideja je daleko od lutanja po sistamima naše galaksije i prepucavanja sa njenim ratobornim stanovnicima Porad toga ona objedinjuje i logičko razmišljanja i pravovremeno reagovanje igrača

Kada se igra učita veći deo ekrana će zauzeti slika komandnog mosta "ENTERPRISE"-a a ispod nje i sa strane sličice sa likovima članova posade I u filmu I n igri svaki član posade ima svoju određenu funkciju pri upravljanju I odbrani broda, pa bi trebalo objasniti svaku ikonu ponaosob (sleva nadesno)

I) Ikona koja nas obaveštava o stanju brodskih motora Postoje dve brzine ili dva načina rada motore WARP i IMPULSE Prva je za putovanja između sistama zvezda u galaksiji a druga za samo putovanja između planeta zvezdanog sistema

II) Ikona sa tri opcije Prva (odozdo nagore) jeste ikona gde podešavamo brzinu broda Postoje dve skale jedna za WARP i druga za IMPULSE SPEED

Druga opcija je karta zvezdanog sistema n kom se trenutno nalazimo

Treća je karta svih sistema galaksija (maksimalno udaljenje je 80 svetlosnih godina) I naziv karte je QUARANTINE ZONE Možemo dobiti I kartu sa sistemima koji su udaljeni 30 sv godina od naše trenutne pozicije (REGIONAL ZONE) i kartu sa sistemima koji su od naše trenutne pozicije udaljeni po 15 sv godina (LOCAL ZONE)

III) „Enterprise" je opskrbljen sa 4 lasera (PHASERS) i 50 torpeda (TORPEDOS) Torpeda su veoma efikasna ali se troše dok su laseri „večni" ali treba pogoditi protivnički brod više puta

Ova ikona sadrži tri opcije

1) Biranje vrsta naoružanja sa kojom otvarate paljbu

2) Aktivira se automatski kada u trećoj ikoni uhvatimo protivnički brod na nišan

3) Predstavlja umanjeni brod okružen sa četiri kvadrata koja imaju funkciju podizanja štita oko broda I okretanja cele slike za 180°

IV) Ovde saznajete sadržaj eventualnih radio poruka koje ste primili Veoma korisna stvar je nabaviti knjigu "ROMULANS CODES FILE"

V) Ikona sa likom Mister Spoka (najpoznatija ličnost filmova iz porodice "STAR TREK") Ovde saznajete podatke o stanju broda (štitovi) I o planeti ili sistemu na koma se trenutno nalazi pokazivač

VI) Trenutno zdravlje svih članova posade možete doznati ovde i to po crti ispod svakog lika (kada nestane taj član posade nestaje iz dalje igre)

VII) Kapetan broda kod koga možete saznati vreme proteklo od početka igre datum u brodskom dnevniku broj uništenih neprijatelja šte se sve nalazi u magacinima broda (STORES) i da li je na moguće transportovanje (TRANSPORTER)

Igranje se otpnlike sastoji u sledećem odaberete ikonu broj dva i opciju sa kartama galaksija (QUARANTINE LOCAL REGIONAL) Zatim odaberete sistem u koji želite ići kliknete na SET COURSE a zatim podesite WARP SPEED Zvučni signal će vam naznačiti da ste stigli u izabrani sistem

Sada treba otići do ikone u kojoj se nalazi karta sistema izabrati željenu planetu I uključiti opciju CONFIRM Ovog puta podesimo IMPULSE SPEED I opet sačekamo zvučni signal

Ako na planeti ima života (kod Spoka označena kao LIFE SUPPORTING) možete se transportovati na nju

U transporter može stati 6 članova posade (sedmi ostaje na brodu) Kada se spustite na površinu planete treba se probiti kroz neku vrstu prolaza a u tome vas sprečavaju razni predmeti (vrata, generator zaštitnog polja pobunjenički komandir) Svaki član posade će imati svoj pametan predlog kako preći ili uništiti prepreku (FIRE PHASER AT IT, DECODE CONTROL LOGIC) Na vama je da od šest ponuđenih odgovora izaberete pravi Ako pogrešite lik će izgubiti snagu i neće moći duže vreme da ga koristite Posle nekoliko prepreka obično naiđete na neki konstan predmet (razmislite šte ćete uraditi ako je jedini pravi odgovor za rešavanje problema kod onog člana posade koji je na brodu)

Igre se završava prerano na dva načina (u većini slučajeva) forsiranje brzine koju motori ne mogu da izdrže (prelazak rastojanja od 80 sv godina sa WARP SPEED 10 itd) ili usled napada neprijatelja (KLINGONS & ROMULANS)

Od prvog se branimo lako što čim dobijamo poruku REDUCE SPEED OR WE'LL BLOW ANY MINUTE smanjimo brzinu a od drugog ulaskom u ikonu broj 3 Prvo odaberemo vrstu oružja a zatim aktiviramo opciju tri I pokazivačem dođemo do broda koji nas napada (u obliku slova T) Tada automatski dolazimo u opciju dva, gde treba dovesti pokazivač do uvećane slike broda u sredini ekrana Tu treba biti vešt i n pravo vreme i na pravom mestu pritisnuti pucanje Zvučni signal označava da je brod uhvaćen u automatski nišan Kada je i to urađeno odšetajte se do bilo kog kvadratića sa strane I otvorite vatru (broj torpeda je 50 — imajte to na umu)

U ovom delu važi pravilo da ako lepo nanišanite za uništenje protivnika treba potrošiti manje torpeda (manje pogodaka lejzerom) nego ako to uradite „ofrlje"

Nekoliko saveta

Pazite se Romulanaca Ponekad će biti miroljubivi i sačekaće da uđete u transporter i da se teleportujete da bi napali nebranjen brod

Imajte uvek na umu da vatru možete otvoriti samo u samoodbrani

Gađajte uvek sa sva četiri lejzera

Razmislite o posetr sledećim sistemima DAKIAK BIXIEM, XYRAM

Nikola Bulatović, Beograd







KILLED UNTIL DEAD

U ulozi najboljeg privatnog detektiva na svetu Harkula Holmsa (Šerlok Holms je već prevaziđen) suočili ste se sa najtežim slučajem u svojoj karijeri Radnja avanture se odigrava u hotelu „Gargoyle" čiji su gosti Mike Stommer Agatha Maypole Lord Peter Flimsey Claudia von Bulow Sydney Meanstreet i vi — Herkus Holms

Sve gore navedene osobe se poznaju a što je vrlo važno jedna od njih planira da ubije drugu Vaš zadatak je da otkrijete koja će osoba pokušati da ubije koju kojim oružjem na kojoj lokaciji i takođe morate otkriti i motiv za ubistvo Za istragu imate ograničeno vreme jer je cilj da preduhitrite ubicu i tako spasete nečiji život.







Na početku će vas kompjuter upitati da li želite da pohađate školu za detektive (School of Sleuthdom) Savetujem vam da odgovorite potvrdno jer je to demo koji vam kasnije u igri može biti od koristi Zatim se morate opredeliti i za jedan od dvadesetak slučajeva koji želite rešiti a koji su vrlo slični.

Nešto o izgledu ove igre Sve vreme sedite u svojoj sobi koja je preuređena u kancelariju i isključivo iz nje vodite istragu (znači — ne smete da se smucate po hotelu vršiti pretrese ili dolaziti u bliske kontakte sa osumnjičenima) Sad se pitate kako voditi istragu E pa lepo pomoću četiri opcija To su sleva nadesno

1 Nadzor (Surveillance) obuhvata 3 nove opcije

— Monitor (skrivene kamere vam omogućavaju da vidite šta se dešava u pojedinim prostorijama)

— Break In (ako odgovorite tačno na postavljeno pitanje iz oblasti kriminalistike policija će vam omogućiti uvid n kartoteku osobe za koju se interesujete)

— Tape (omogućava prisluškivanje tuđih razgovora)

2 Telefon (jasno je čemu služi)

3 Notes (sadrži sve podatke koje ste saznali)

4 Arhiva („Files" — engl) sadrži kratke biografije svih osoba Kada razgovarate sa nekom osobom preko telefona, u gornjem delu ekrana pojaviće se lice te osobe dok će se n donjem delu ekrana pojaviti mnoštvo pitanja od kojih ćete džojstikom izabrati jedno Da bi vaš posao bio što teži niko neće odgovarati konkretno na vaša pitanja (na primer ako pitate nekoga da li Agata može biti ubica može vam odgovoriti „Ako je Agata ubica zločin se neće dogoditi u biblioteci") No nije sve ni tako crno, jer vi vidite lice osobe sa kojom pričate pa na osnovu njenog izraza lica dok vam odgovara na postavljeno pitanje možete mnogo toga zaključiti

Tako bi glasio i savet broj jedan ako rešavate lakši slučaj oslanjajte se na reakcije osobe sa kojom pričate Ako ste se odlučili da rešite teži slučaj — maksimalno se koristite opcijom „Nadzor" i obratite pažnju na anonimne telefonske pozive Drugi savet je — nikada ne optužujte nekoga ako niste potpuno sigurni u sakupljene dokaze (ako optužujete nevinu osobu ubica će vas staviti na nišan i tada vam preostaje samo da isključite računar i počnete od početka)

Treći, i ujedno i poslednji savet je obavezno se „naoružajte" rečnikom (avantura je pisana na engleskom jeziku) jer u protivnom svaki pojedinačni slučaj ne biste rešili za manje od dva časa

Branislav Jovanović, Bor

Sve o igri

CARDS



CARDS je program koji u sebi sadrži pet igara sa kartama. To je jedna od retkih igara koja pruža mogućnost izlaska iz programa bez resetovanja računara. Igra se startuju klikom na njihovo ime u meniju ili pritiskom na odgovarajući funkcijski taster. Postoji i meni za opcije u vezi sa nekim igrama pa će njegove stavke biti opisane sa igrom s kojom su u vezi. Krenimo redom.

Blackjack je igra veoma slična kod nas poznatoj igri ajnc. Kada je startujete prikazaće se tabela u koju unosite imena, ukupan iznos novca i ulog za sledeće deljenje za četiri igrača. Naime ovde mogu igrati do četiri igrača protiv kompjutera. Kada ste sve uneli pritiskom na RETURN počinje deljenje. Kompjuter deli svakom igraču po dve karte (on uvek deli) i sve su karte otvorene osim njegove druge karte. Cilj vam je da imate zbir što bliži ili jednak 21 ili da imate pet karata čiji je zbir manji od 21. Ako pređete 21 ispadate iz tog kruga, „Jipoe" ste. U Blackjack-u ne postoji takozvani „suvi" ajnc kada imate dva keca već tada imate zbir 2 ili 12 jer se kec može brojati kao 1 ili 11. Dakle ajnc (blackjack) predstavlja par jednog keca i neke slike (karte veće od devetke). Kada dobijate blackjack automatski ste dobili deliocu iznos novca koji imate se povećava za 150% uloga, a ostali igrači nastavljaju da igraju taj krug. U toku igre imate četiri mogućnosti koje deliocu saopštavate pritiskom na tastere 1, 2, 3 i 4. To su: Hit — „gađaj" me sa još jednom kartom. Stand — dosta. Double — ovim udvostručujete svoj ulog u tekućem krugu i dobijate još jednu kartu i nemate više mogućnosti dupliranja uloge. Separate — ovu mogućnost imate na raspolaganju ako su prve dve karte koje ste dobili iste. Na primer dva kralja, dve trojke itd. Pritiskom na 4 ćete razdvojiti te dve karte i nastavljate da igrate na dva talona koja su sasvim nezavisna. Svoj red završavate pritiskom na 2 kada procenite da Vam zbir odgovara, kada ste tropa ili kada imate pet karata a niste tropa. Kada svi igrači završe kompjuter pokušava da vas „pronađe". Ako uspe (ima veći ili jednak zbir) gubite ulog u suprotnom ga dobijate.

Cribbage je igra u kojoj igrate protiv kompjutera. Pobednik je onaj ko prvi stigne do 121. Počinje tako što se od šest karata izdvoje dve koje se odvajaju na stranu. Vaše dve karte i one dve koje kompjuter izdvoji čine grupu pomoću koje će igrač koji je podelio karte imati mogućnost da dobije dodatne poene. Kartu koju izdvajate birate pritiskom na tastere 1, 2, 3, 4, 5 ili 6. Kada ste izdvojili dve karte biće presečen špil i okrenuta jedna karta koja povećava broj kombinacija. Zatim naizmenično izbacujete karte i kompjuter trudeći se da spojite neku od kombinacija koje donose poene. Kombinacije i broj poena koje donose su sledeće:

zbir karata na talonu je 31 i 15 — 2 poena
zbir karata na talonu je 31 posle predaje poteza — 1 poen
predaja poteza drugom igraču donosi mu — 1 poen
igrač koji postavi zadnju kartu na talon dobija — 1 poen
par — 2 poena
tri iste karte — 6 poena
četiri iste karte — 12 poena
fleš (četiri Vaše karte u istom znaku) — 4 poena
fleš (četiri Vaše i peta zajednička karta u istom znaku) — 5 poena
run (zadnjih 3—8 karata čine niz bez obzira na redosled) — 3—8 poena
pub u istom znaku kao zajednička karta — 1 poen

Zbir na talonu ne može da pređe 31. Ako nemate kartu kojom možete da odgovorite predajete potez drugom igraču pritiskom na RETURN i on dobija jedan poen. Ukoliko ni on nema kartu kojom može da odgovori karte na talonu se okreću, zbir se vraća na nulu i v nastavljate igru ali ne dobijate nikakve poene. Kada svih osam karata koje su u igri budu na talonu igrač koji je stavio poslednju kartu dobija jedan poen. Sada kompjuter pronalazi sve moguće kombinacije koje donose poene među vašim kartama uzimajući u obzir i dodatnu zajedničku kartu. To isto čini i sa svojim kartama. Zatim to isto čini i sa one dve vaše i dve njegove karte koje su izdvojene na početku deljenja i poene sabira onome koji je u tom krugu podelio karte.

Klondike je jedna od mnogobrojnih vrsta pasijansa. Na talon se postavi 36 karata u osam redova, tako što je prva karta u stupcu okrenuta licem na gore a ostalih 16 se stave na stranu. Karte se ređaju jedna na drugu po pravilima: crvena karta može na crnu a crna na crvenu (karo i herc mogu na pik ili na tref a pik i tref mogu na karo ili herc) i samo manja karta po vrednosti može na veću (kec je najmanji a kralj najveći). Cilj igre je da složite sve karte po znaku od keca do kralja. Svake složene karte vam donose pet dolara. Karte slažete tako što kliknete na kartu koja je na redu da bude složena. Karte se moraju slagati po redu tako da ne možete da izbacite trojku a da niste pre toga izbacili prvo keca pa dvojku u tom znaku. Takođe ne možete da izbacite kartu koja nije na vrhu gomile na talonu. Igrate tako što na talonu slažete karte jednu na drugu da bi otvorili one karte koje se na vr... To radite tako što kliknete na g...u koju želite da pomerite i drž... dugme na mišu i vučete gomilu do mesta na koje želite da je premestite. Na prazno mesto na talonu možete da postavite samo kralja. Kada niste više u mogućnosti da premeštate i izbacujete karte služite se ostatkom karata sa leva strane. Kliknite na tu gomilu i jedna karta se okrene tako da možete da je vidite. Tada možete da se služite s njom kao i sa kartama na talonu. Kada ne možete ništa više da učinite ili kada vam se pasijans „otvori" završavate igru. Svako novo deljenje vam oduzima 52 dolara a svoj skor brišete pomoću Clear Total (F8).

Poker Squares igrate slažući karte na talon u kvadrat i to u pet redova po pet karata u svaki. Kompjuter neprestano meša karte a vi ga zaustavljate klikom ili pritiskom na neki taster. Kartu na kojoj ste prekinuli mešanje postavljate na talon klikom na željenu poziciju. Kako koju kartu postavite na talon kompjuter sabira skor koji ste tim postavljanjem postigli. Kombinacije koje donose poene su:

par 2 poena
dva para 5 poena
triling 10 poena
kenta 15 poena

boja 20 poena
ful 25 poena
poker 50 poena
fleš 75 poena

Kao što vidite u igri ne postoji fleš rojal. Poeni se dobijaju za kombinacije složene u svakom redu i svakom stupcu. Postupajte pažljivo sa opcijama u vezi sa ovom igrom a to su Save i Clear high score jer one rade direktno sa diskom. Tako će opcija Clear high score obrisati stari high score sa diska a Save high score snimi ti high score preko starog.

Solitaire je igra koja zahteva dosta razmišljanja i kombinatorike. Pred sobom imate sve 52 karte poređane u grupe po tri i jednu izdvojenu kartu. Cilj igre je da sve karte složite po redu iz tri deljenja. Karte pomerate i izbacujete klikom na njih. Samo gornje karte u svakoj grupi možete da pomerate i slažete jer su složene jedna preko druge. Tako na primer ako je kralj u piku složen preko keca u piku u tom deljenju sigurno nećete početi da slažete pike jer kralj ne može da se premešta pošto je on najveća karta. Naime i u ovoj igri važi pravilo da samo manja karta može da se stavi na veću. Ako ste preko neke karte već nanizali određen broj karata toj karti ne možete pristupiti dok ne složite ceo niz karata koji je preko nje. Kada ste iscrpeli sve mogućnosti pritiskom na F9 ili biranjem opcije 1st redeal karte koje nisu složene će se izmešati i ponovo raspodeliti u grupe a kada i tu iscrpite sve mogućnosti poslednju šansu ćete dobiti pritiskom na F10 ili opcijom 2nd redeal. Grupe od četiri karte koje je poslednje od svih na ekranu je mesto na koje se slažu karte. One vam govore dokle ste stigli u slaganju karata.

Milan Adamov, Beograd

Moja top lista

1. Thing
2. Train
3. Redweed
4. Beyond the Ice Palace
5. Piggy
6. Pacmania
7. Football manager 2
8. Match day 2
9. Ghosts n Goblins
10. Flintstones

Lukijan Popović, Beograd

Moja top lista

1. Basket master
2. The train
3. Out run
4. Match day 2
5. Track suit manager
6. Elite
7. TT racer
8. Football manager
9. Roy of the Rovers
10. Tetris

Mario Žamić, Brala

Sveže učitano

Sinclair

MANSTERI
(THE MUNSTERS)

Arkadna evantura?

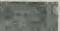
U principu da šta bi rekli na Radio Jarevanu u praksi je to rezume se šunjanje po raznim prostorijama u zamku Manstarovih skupljanje svakojakih predmeta i pucanje na raznovrsna anđama i karakondžula

Sve to da biste očistili zamak od zlih duhova ili letećih đavola ka, ili već tako nečeg Imate samo jedan život ali energiju možete obnavljati (s malo sreće naievno) a imate i moć bacanja čini Ovo drugo shvatite sasvim doslovno čini imaju oblik lopte i bacate ih na neprijatelje neki put je dovoljan jedan pogodak a naki put — kod okorelijih — treba i više

U svakom slučaju izbegavajte kontakt s neprijateljima ako možete Kako vas koji dotakne tako — po starom principu kompjuterskih igara — gubite nešto energi je

Sve je to već viđeno ovde ili onde ali šta ćete prošla su ona vremena kad je svaka druga igra bila nešto novo Sad možemo biti srećni ako je svaka stota delimično nova

ZVEZDANA FARSA
(STAR FARCE)



Ako ste igrali STAR FORCE može vam se učiniti da je STAR FARCE neki štos neka igra reči neka parodija

Bićete u pravu ali samo što se naslova tiče Igra je sasvim ozbiljna naročito su ozbiljni vanzemalj ci čija je jedina svrha da vas ubiju (ili bar da pokušaju)

Vi i bez toga imate muka jer programer je hteo da napravi šarenu pozadinu a znamo šta se na „Sinkleru" dešava kad se dve boje sudare s trećom No pošto vam je sve vreme glava u torbi možda i nećete obraćati pažnju na ono što kritičari zovu likovnim kvalitetom Gledaćete pre svega da uhvatite kakav dodatak za svoju letelicu — nešto što će vam udvostručiti vatrenu moć ili vas štititi od prvog pogotka — a nastojaćete naravno i da eliminišete što više neprijatelja. Možete ako se jednog nivoa zasitite propadati kroz rupe u druge nivoe ali ništa vas neće spasiti susreta sa matičnim brodom jer tu grdosiju morate uništiti ako imate imalo želje da pređete u neki viši nivo No to ste i ranije znali to spada u opšta mesta pucačkih igara

L.E.D. OLUJA
(LED STORM)

LED diode su ako ne grešim one diode koje svetle Njih ovde nema ali nište za to nema ni olu je Sem ako reč „oluja" ne shvatimo kao hiperbolu kao automobili su ovde brzi poput oluja ili već tako nekako

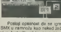
Automobili? Da Ovo je trkačka igra a od vas se u njoj traži samo jedno da stignete od cilja pre no što izgubite svu energiju Na drumu drugim rečima nema direktne opasnosti po vaš život Možete propasti kroz rupu možete čak i tresnuti svom snagom u neku prepreku — a opet ostati živi (i dobiti nova kola) Jedino što ćete izgubiti biće energija i život će vam otići samo ako je u trenutku nesreće niste imali dovoljno

Naravno energija se obnavlja kad pregazite kantu sa gorivom ili kad prošišate kroz kontrolni punkt

Drum je veoma nezgodan pa ćete čas ubrzavati a čas usporavati čas motati ulevo a čas skretati udesno — ali možete i skakati ako hoćete da preskočite rupu ili naskočite na tuđ auto i uništite ga dovoljno je samo da pritisnete FIRE

I tako u devet nivoa sve jedan sličniji drugome

BMX SLOBODNIM STILOM
(BMX FREESTYLE)



Postoji opasnost da se igra o BMX u namnoži kao nekad začevi u Australiji ali to naravno ne znači da svaka mora biti za in minus Ima i zabavnih

Ovde ne primer kušate svoje umeće i spretnost u šest disciplina, takmičeći se protiv kompjutera ili živog partnera kako vam se više sviđa

Na početku vozite kaskaderski na zadnjem točku trudeći se da stignete što dalje pre no što se prevrnete Potom morate pr...ka-ti rampu (i ostati u sedlu k... i je preskočite) a u trećoj disciplini vozite levo-desno po čuvenoj presečenoj cevi to jest po pisti koja u preseku izgleda kao donja polovina slova O Kad to apsolvirate dolazite na stazu na kojoj treba izvesti što više vozačkih majstorija ne bste li izbegli gomilu kvarnih prepreka Sledi tzv sporа trka naime trka u kojoj pobeđujete ako do cilja stignete poslednji Najzad kad ste sa već dobio preznojili čeka vas skakanje sa četvrt-cevi zaletite se svom snagom skočite sa ivice i nastojte da u odskoku na upropastite stvar

Posle toga vadite iz ormana žutu majicu i navlačite je

STARINSKE BITKE
(ANCIENT BATTLES)

Najbolji način da se ratuje ni ko ne gine nikome ne gori kuća na kraju igre svi živi i zdravi

Na početku biranja Hoćete li bitku iz kompjuterske memorije ili biste radije da sami našto smisli te? Pa onda hoćete li igrati protiv kompjutera ili protiv jednog prijatelja ili protiv dvojice prijatelja? I tako dalje Imate štaviše i varijantu u kojoj vaš verni „Sinkler" uzima na sebe poslove nižih starešina i sprovodi vaša opšta uputstva

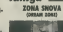
Možda je najveće zadovoljstvo smisliti bitku po svome U tom slučaju prvo birate vojsku koju ćete baciti u vatru pa onda projektujete bojište — onde potok onde cvet što bi rekao J J Zmaj — pa onda ko živ ko mrtav

Ako biste radije neku gotovu bitku stvari će trajati kraće no u svakom slučaju morate paziti na raspored jedinica kondiciju vojnika visinu borbenog morala, transport snabdevanje i treće drugih stvari

Ako niste profesionalni oficiri neće vam biti lako

Amiga

ZONA SNOVA
(DREAM ZONE)



ili tačnija zona košmara U ovoj grafičkoj avanturi vi kao sanjate da ste se našli u gradu birokratizovanom do krajnje mere i da ne znate kako ćete se snaći

Prvi vam je problem ući u kafanu Bez propusnice ne može Pojavljuje se čovek koji prodaje propusnice ali kupovinu ne možete obaviti ako nemate formular Ministarstva informacija a formulari se izdaju u sobi toj i toj ali kad do te sobe najzad stignete kažu vam da morate ići u drugu i tako dalje ako ste ikad u stvarnom životu imali posla u opštini, jasno vam je kako se u ovoj igri stvari razvijaju

Ako uspete, idete dalje u luna park u podvodnu pećinu, u orijentalnu palatu i ko zna kuda sve još Sve vreme kuburite sa svakojakim problemima teško rešivim iako imate mapu a povremeno dobijate i pokoju uputstvo Sve komande možete ukucavati načinom klasičnim ali neke možete izdavati i drugačije pomoću miša i ikona Valja znati još i ovo scene jave urađene su crno-belo i nejasno a scene sna u boji i izoštreno Najzanimljivije stvari ipak će vam se dešavati na trećem mestu među javom i med snom

ŠAH NA ODREĐENO VREME '89.
(CLOCK CHESS 89)

Ovo nije igra nego robna kuće nema šta nema

Ako ste početnik možete sebi dodeliti više vremena za razmišljanje nego kompjuteru Ako ste dobar igrač možete mu dati koliko i sebi Možete dalje odrediti da i vi i on morate povući toliko i toliko poteza za toliko i toliko vremena Ako biste radije smišljali probleme nakrivite kapu postoji opcija i za to kao što postoji i za analizu pozicija i za igru naslepo i za igru bez vašeg učešća u njoj kompjuter igra sam protiv sebe što je poznato kao primenjena šizofrenija

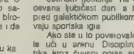
Pre nego što išta od toga uradite, valja vam birati između dvodimenzionalne i trodimenzionalne varijante

U igri možete izvoditi i neke dodatne burgije ako samo zaželite kompjuter će vratiti potez ili promeniti mesto s vama, ili čak pokazati vam koje poteze smišlja

Ako i posle toga ne pobedite niko vam neće biti kriv

Atari ST

LJUBIČASTI SATURNOV DAN
(PURPLE SATURN DAY)



Jednom godišnje na Saturnu osvane ljubičast dan a tada se pred galaktičkom publikom održavaju sportske igre

Ako ste u to poverovali možete ući u arenu Disciplina prva trka kroz čuveni pojas asteroida oko Saturna Ne bi bila laka ni bez asteroida uz njihovu pomoć uma da bude i pogibeljna. Disciplina druga pucanje na kuglu energije i skupljanje krhotina te iste kugle pošto je pogodite i razbijete Ni ovde nećete imati mira namučiće vas suparnici i neprijateljski nastrojen predeo Disciplina treća reaktiviranje jedne polovine mozga Tu vodite kuglicu koja se načinom fliperskim odbija od površine mozga i uključuje prolaze ali vaš protivnik koji radi to isto na drugoj polovini sabotira vaše poduhvate (Utešite se podatkom da možete i vi sabotirati njega) Disciplina četvrta gola pucnjava Nebeska tela lete kao glineni golubovi ili već tako nekako a vi treba da pogodite što više njih da biste stekli dovoljno energije za prelazak u viši stepen takmičenja To obavite triput posle čega ulazite u spektakularan finale Ako pobedite dobijate vruć poljubac od Mis Saturnovog ljubičastog dana i vraćate se kući

Lokalne računarske mreže i distribuirane baze podataka

Mreže u fokusu

Intenzivan razvoj lokalnih računarskih mreža započeo je 1980. godine kada su kompanije „Xerox", DEC i „Intel" uvele Ethernet specifikaciju kao industrijski standard. „Ethernet" je danas jedna od najpoznatijih lokalnih mreža, a razvijena je krajem sedamdesetih godina u „Xerox Palo Alto" istraživačkom centru za potrebe istraživača koji su većinu svojih zadataka izvršavali na namenski korišćenim mini računarima. Istraživači nisu imali potrebe da komuniciraju sa nekim host sistemom, već su želeli da između sebe razmenjuju različite informacije, uključujući tekstualne poruke (elektronsku poštu), datoteke i programe. Takođe je postojala potreba za korišćenjem uređaja kao što su laserski štampači i za povezivanjem sa telefonskom mrežom. Iz ovakvih zahteva proistekla je potreba za razvojem Ethernet mreže.

Sa razvojem personalnih računara, lokalne računarske mreže dobijaju na značaju. Neinteligentni terminali povezani na mini i host računare počinju da se zamenjuju sa inteligentnim radnim stanicama, a ove se dalje međusobno povezuju formirajući kompleksne informacione sisteme. Sistemi za upravljanje bazama podataka (DBMS) koji se koriste na personalnim računarima po svojim karakteristikama odgovaraju onim koji se koriste na velikim sistemima. Kod mnogih informacionih sistema distribuirane baze podataka predstavljaju atraktivnu alternativu u odnosu na centralizovane sisteme, nudeći povećanu raspoloživost podataka, povećanu pouzdanost i bolja vremena odziva. Razvoj lokalnih mreža poslednjih godina značajno je smanjio vremena potrebna za komuniciranje, što je predstavljalo usko grlo kod računarskih mreža, tako da je omogućena praktična primena distribuiranih baza podataka.

Distribuirane konfiguracije informacionih sistema zasnovane na korišćenju lokalnih računarskih mreža i distribuiranih baza podataka uklapaju se prirodno u hijerarhijsku organizacionu strukturu mnogih savremenih organizacija. Ove strukture u obliku stabla uobičajeno se sastoje od funkcionalno ili operativno posebnih jedinica sa različitim stepenima autonomije (grupe, odeljenja, radne jedinice itd). Takve jedinice često kreiraju i održavaju svoje sopstvene operativne podatke. U takvim uslovima zahteva se pristup lokalnim informacijama, dok su izvori podataka distribuirani u ograničenoj geografskoj oblasti. Na bilo kom nivou hijerarhije u organizaciji potrebno je obezbediti efikasan mehanizam koji omogućava ne samo pristup lokalnim podacima, već i podacima koji pripadaju jedinicama na nižim nivoima.

Većina baza podataka koje su danas u upotrebi su centralizovane, to jest podaci su smešteni samo na jednom računaru. U okviru jedne organizacije često postoji više centralizovanih baza podataka smeštenih na različitim računarima, na šta obično utiču sledeći razlozi:

- organizacija je geografski distribuirana,
- ukupni informacioni zahtevi su suviše veliki da bi mogli da se održavaju na jednoj bazi podataka,
- više baza podataka je razvijano iz različitih razloga, posebno za razne delove organizacije,
- uvođenje personalnih računara u organizaciju dovelo je do formiranja više malih baza podataka.

Mnoge organizacije imaju, međutim, globalnu potrebu za brzim i pouzdanim pristupom podacima koji se nalaze u ovim posebnim bazama. Danas je ovo još uvek komplikovano i korisnik mora da prođe sledeće korake da bi to realizovao:

- određivanje računara koji sadrže podatke kojima treba da se pristupi,
- formulisanje više upita koji se izvršavaju na različitim računarima,
- prenos rezultata na jedan računar da bi se integrisali,
- integracija rezultata,
- izdvajanje odgovora na postavljeno pitanje iz integrisanih podataka.

Ovi zadaci su komplikovani, zahtevaju puno vremena i podložni su greškama. Njihova automatizacija postiže se primenom distribuiranih sistema za upravljanje bazama podataka (DDBMS). Distribuirani DBMS omogućavaju administratoru baze podataka da izvrši raspodelu integrisane baze podataka i njenog upravljačkog sistema i da ih distribuira u više autonomnih geografski udaljenih čvorova. To se vrši na takav način da je lokacija bilo kog posebnog skupa podataka transparentna i za korisnike i za aplikacije.

Prednosti distribuiranih DBMS su višestruke. Prvo, distribuirani projekt odražava organizacionu strukturu. Drugo, omogućava se da podaci budu smešteni lokalno, pod lokalnom kontrolom. Lokalno smeštanje podataka smanjuje vremena odziva i cenu komunikacija, a povećava raspoloživost podataka. Treće, raspodelom podataka po višestrukim autonomnim lokacijama, efekt ispada sistema iz rada lokalizuje se na mesto nastanka, podaci smešteni na ostalim računarima i dalje su raspoloživi.

Četvrto, veličina baza podataka i broj korisnika ne moraju da budu limitirani memorijskim i procesnim kapacitetima. Distribuirani DBMS radi sa praktično neograničenim brojem računara, diskova i korisnika. Ovo ima za posledicu peru prednost. U mnogim slučajevima, višestruki integrisani mali sistemi mogu da koštaju znatno manje od jednog velikog sistema. Šesto i možda najvažnije, distribuirana baza ne mora da bude ograničena fizičkom organizacijom podataka. Distribuirani DBMS i rečnik podataka mogu da se iskoriste za integraciju, pristup i ažuriranje heterogenih baza podataka, koje se za korisnika javljaju kao jedna logička baza podataka.

Sve šira primena personalnih računara, kao i sve moćnije mašine iz ove klase sistema koje se javljaju iz dana u dan, kombinovane sa softverskim rešenjima koja olakšavaju distribuiranu obradu, doveli su do novih arhitektura u ovoj oblasti.

STRATEGIJE ZA POVEZIVANJE RELACIONIH DBMS ZASNOVANIH NA PERSONALNIM RAČUNARIMA

Jedan od ključnih ciljeva svih velikih proizvođača računarske opreme je da integrišu PC računare, miniračunare i velike sisteme (mainframes) u jedinstvenu informacionu mrežu. Takav primer je i arhitekture koju razvija IBM Systems Application Architecture (SAA), na čijoj realizaciji se radi, ali su specifični proizvodi još uvek u različitim stepenima razvoja. U međuvremenu, veći broj nezavisnih kompanija razvija softver za rad u uslovima distribuirane obrade, u najvećem broju slučajeva zasnovan na SQL (Structured Query Language) jeziku za relacione sisteme za upravljanje bazama podataka (RDBMS).

SQL je formalni jezik za specifikaciju uobičajenih operacija sa bazom podataka (na primer: pretraživanje zapisa, dodavanje, izmena i uklanjanje zapisa). IBM je deklarisao SQL kao SAA standard, a ovaj je jezik takođe standardizovan od strane međunarodnih organizacija (ANSI SQL i drugi). IBM trenutno koristi SQL sa bazom podataka DB2 (na velikim mašinama) i SQL/DS (na srednjim sistemima), a planiran je SQL DBMS sa OS2/Extended Edition (na PS/2 sistemima). Većina proizvođača opreme takođe podržava neki oblik SQL DBMS za svoje proizvode.

SQL DBMS se uobičajeno sastoji od dva dela:

a) SQL „mašina", koja može da prihvata SQL zahteve iz aplikativnog programa napisanog na nekom od standardnih programskih jezika i
b) skupa pomoćnih programskih modula (kao što su interaktivni upitni jezik, generator izveštaja, generator formi i jezik četvrte generacije) koji omogućavaju da se pristupi mašini, a da se ne mora koristiti konvencionalni programski jezik.

Značaj SQL-a kao standarda je što on nudi način za potencijalni razvoj aplikacija (i pomoćnih modula) nezavisnih od specifičnih DBMS mašina kojima pristupaju. Aplikativni programi koriste zajednički skup komandi kojima se zahtevaju operacije sa bazom podataka, a da ne moraju da vode računa o mehanizmu kako se ove komande obrađuju na SQL mašini koja je u pitanju. Koristeći SQL, programeri mogu da kreiraju aplikacije sa mogućnostima kooperativne obrade, tako što se korisnički interfejs nalazi na personalnom računaru, ali se podaci nalaze na PC serveru, miniračunaru ili na host sistemu.

Distribuirani SQL na PC preuzet od mini računara

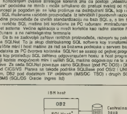
Danas postoje dve osnovne strategije u ponudi distribuiranih SQL sistema na PC računarima. Prva strategija je podržana od firmi koje nude distribuirane verzije DBMS sistema preuzete od uspešno razvijenih centralizovanih verzija sa miniračunare. Primer ovih sistema su Oracle Corporation (SQL*Star) i Relational Technology (Ingres*Star). Oracle je jedan od prvih DBMS koji su počeli da koriste naprednu tehnologiju zasnovanu na 32 bitnim sistemima, kao što su Compaq 386 i IBM Personal System/2. Ovi proizvođači takođe nameravaju da podržavaju DBMS sisteme drugih proizvođača, kao što je IBM-ov DB2. Međutim, u praksi njihov naglasak je na obezbeđenju homogene sredine za rad sa bazom podataka, u kojoj se prilagođene verzije njihovih proizvoda izvršavaju na PC računarima, mini i host sistemima i gde je obezbeđen pogled na sistem sa jedinstvenom bazom podataka.

LAN verzije distribuiranog SQL-a

Još jedna strategija za distribuirani SQL zasnovana je na korišćenju PC LAN-ova. Primer distribuiranog SQL proizvoda specijalno razvijenog za PC LAN-ova je SQLBase firme „Gupta Technologies" SQLBase transformiše jedan ili više PC/AT ili PC/386 mašina u PC server baza podataka na bilo kom NetBios kompatibilnom LAN u Aplikativni programi (ili front-end moduli) izvršavaju se na individualnim PC sistemima a mogu simultano da pristupe (bilo da postavljaju upit ili da vrše ažuriranje) višestrukim serverima baza podataka na mreži s time da ne moraju da znaju fizičku lokaciju servera

Korišćenjem ovakve strategije korisnik zna da postoje „višestruke baze" podataka na mreži i može simultano da pristupi svakoj od njih Koncept je pogodan jer se lako proširuje na distribuirani SQL, koji radi sa SQL mašinama različitih proizvođača Iz tehničkih i poslovnih razloga većina proizvođača će izvršiti standardizaciju na bazi SQL a, s tim što će različita SQL mašina biti korišćena za PC računare miniračunare i host sistema Većina aplikacija u mreži koristiće kao radne stanice PC računare a ne neinteligentne terminale

Da bi se zadovoljili zahtevi različitih proizvođača, razvijeni su paketi tipa SQLNet To je skup distribuiranog SQL softvera koji transformiše različite mini i host mašine za rad sa bazama podataka u servere baza podataka za PC čvorova korisnika SQLNet se sastoji od gejtvej programa koji prosleđuju SQL zahteva odgovarajućem hostu a host programi koji koriste mogućnosti mini i velikih SQL mašina odgovaraju na te zahteve Za sada SQLNet povezuje samo SQLBase (pod PC DOS) i DB2 pod CICS/VS Planira se takođe podrška SQLBase pod OS/2 i UNIX-om, DB2 pod dodatnom TP okolinom (IMS/DC TSO) i drugih SQL DBMS (SQL/DS Oracle Ingres itd)

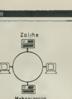


Sl 1 Distribuirani SQL DBMS

U poslednjih nekoliko godina pokazalo se da je glavna prednost SQL e da razdvaja front-end oruđa (i aplikacije) od SQL mašina PC računari na kojima se nalazi front-end kôd počinju da zamenjuju neinteligentne terminale Relacioni sistemi postaju najviše korišćeni u produkciono orijentisanim aplikacijama Kako većina firmi koja podržavaju produkciono orijentisana aplikacija zahteva pristup bazama podataka na hostu koji mogu da koriste drugu SQL mašinu od one koja se koristi za lokalnu bazu podataka, rešenja se više mašina je poželjno Takođe se sve širim korišćenjem lokalnih računarskih mreža javlja se potreba za distribuiranim DBMS sistemima koji su optimizirani za rad sa LAN-ovima **Na slici 1** dat je primer organizacija koja koristi distribuirani SQL DBMS se više mašina Ovde se SQLBase koristi da obezbedi pristup lokalnoj bazi podataka na personalnim računarima, a u isto vreme SQLNet obezbeđuje efikasan pristup podacima n centralnoj bazi koji se čuvaju na IBM host sistemu Za operatora koji unosi podatke ili za korisnika na lokalnoj mreži koji postavlja upite sačuvan je pogled na jedinstvenu bazu podataka (pisko front-end aplikacija)

KORIŠĆENJE DBMS U LOKALNIM RAČUNARSKIM MREŽAMA SA PC RAČUNARIMA

Eksplozija primene personalnih računara, od najmanjih od najvećih radnih organizacija odvela je do potrebe da se skupi uređaji kao diskovi i štampači dele između korisnika ovih računara Korisnici u okviru manjih organizacionih celina imaju potrebu da međusobno raspodeljuju podatke



Sl 2 Lokaciono nezavisne BP

i uviđaju da to može da se radi na višem stepenu automatizacije nego što je razmena disketa ili papirnih kopija Isti korisnici takođe počinju da zahtevaju pristup selektovanim podacima koji se nalaze u centralnim bazama podataka da bi ih preneli n svoje lokalne baze i dalje koristili Pojedinačna rešenja za gore navedene zahteve počela su da se javljaju tokom poslednjih godina Međutim skorašnja najava od strana IBM-a u oblasti SNA (Systems Network Architecture) i lokalnih računarskih mreža otkrile su obimniju dugoročnu strategiju za povezivanje personalnih računara miniračunara i host mašina Ova strategija se kreće oko softverske povezanosti preko komunikacionog protokola APPC (Advanced Program to Program Communications Protocol) poznatog takođe kao LU 6 2 od fizičke povezanosti preko IBM-ovog Token Ring mrežnog hardvera

U ovom procesu razvoja komunikacija i LAN-ova uloga DBMS sistema je u stanju promena Proizvođači postojećih DBMS sistema brzo uvode „mrežna" verzija svojih proizvoda Ovi novi proizvodi imaće širokog uticaja na raspodelu resursa u složenim organizacijama Sa opštim trendom zamene neinteligentnih terminala sa radnim stanicama zasnovanim na PC računarima aplikativni softver će se sve više pomerati ka nivou radnih stanica

Da bi DBMS sistem koji se koristi u LAN-u bio efikasan mora da zadovolji sledeće kriterijume

- mora da obezbedi pristup podacima na mreži nezavisan od lokacija
- čvor na mreži u koma se nalazi server baza podataka mora da bude raspoloživ i za druga aplikacije
- mora da obezbedi bezbedan rad sa raspodeljenim bazama podataka
- mora da optimizira performanse pristupa bazi podataka od strane više korisnika
- potrebno je da automatski osigura konzistentnost raspodeljene baze podataka i obezbedi različite nivoe konzistentnosti, uključujući i fizičku i logičku konzistentnost
- mora da detektuje otkaz bilo koje komponente mreža i da sadrži mehanizam za automatsko obnavljanje iz takvog otkaza
- mora da ponudi aplikativna oruđa koja maksimiziraju korišćenje interfejsa korisnika sa inteligentnim radnim stanicama

U daljem tekstu sledi objašnjenje svakog od gornjih zahteva i nekih faktora projektovanja koji određuju koliko dobro LAN DBMS ispunjava ove zahteve

Nezavisnost lokacije

Kao što je ranije rečeno ključni razlog da se izabere LAN DBMS umesto tradicionalnog multi-korisničkog DBMS-a je što se pristup bazi podataka zahteva sa PC radnih stanica a ne sa neinteligentnih terminala To znači da će se podaci nalaziti u jednom ili više čvorova sa serverima baza podataka i da će se aplikativni programi koji pristupaju podacima nalaziti u pojedinim čvorovima korisnika Nezavisnost lokacije znači da aplikativni program mora da bude u stanju da pristupi bilo kojoj stavki podataka u mreži a da ne mora da specifikuje njenu fizičku adresu **Na slici 2** ilustrovana je mreža lokaciono nezavisnih baza podataka kojima se pristupa sa višestrukih aplikativnih čvorova

U cilju postizanja nezavisnosti lokacija LAN DBMS mora da zna gde su locirane različite baze podataka (to jest mora da ima rečnik distribuiranih podataka) Izbor ko održava ovo znanje (mrežni nivo nivo sistema datoteka ili DBMS nivo) i koji tip informacija se održava utiče na nivo lokaciona nezavisnosti i performanse LAN DBMS sistema.

Čvor sa serverom baze podataka koji služi i kao radna stanica

Podaci u bazi podataka mogu da budu smešteni kao skup fizički razdvojenih datoteka ili mogu da budu u jednoj datoteci koje sadrži više međusobno povezanih tabela Bez obzira na način smeštanja podataka poželjno je da funkcije servera baze podataka ne obavlja samo jedan fizički čvor Glavni razlog za to je da se omogući da server radi i kao

radna stanica. Drugi razlog (mada je to nešto teži cilj koji treba postići) je da se organizuje smeštanje podataka na takav način da otkaz pojedinog servera baza podataka ne dovede u pitanje rad celog sistema. Treći razlog (koji će se razmatrati kada se govori o performansama) je da se omogući pristup pojedinom korisniku bez negativnog uticaja na performanse.

Bezbednost rada

Svaki multikorisnički DBMS treba da obezbedi odgovarajuće zaštite da bi se osiguralo selektivan pristup raspodeljenoj bazi podataka. U praksi to uobičajeno znači da svaki korisnik na svom PC računaru mora da specifikuje ime i lozinku da bi se povezao sa bazom podataka kao i da mora da postoji neka shema za autorizaciju privilegija koja za svakog korisnika definiše koja operacije može da izvršava. U tabeli 1 navedene su tipične privilegije koja mogu da se dodele korisniku LAN DBMS-a. Za privatne baze podataka (to jest za one kojima pristupa samo kreator baze) mora da postoji način da se baza definiše kao privatna tako da nije potrebno ispunjavati sve zahteve bezbednosti.

1 Kreiranje (tabela, indeksa, kolona)
2 Izmena (tabela, indeksa, kolona)
3 Uklanjanje (tabela, indeksa, kolona)
4 Selekcija (redova, kolona)
5 Ažuriranje (redova, kolona)
6 Unošenje (redova)
7 Uklanjanje (redova)

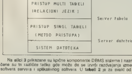
Tabela 1 Tipične privilegije baze podataka

LAN DBMS namenjen za korišćenje kao server u manjim organizacionim celinama ili kao način za pristup centralnoj bazi podataka treba takođe da obezbedi dodatne mere bezbednosti. Ona treba da uključe mogućnost dodeljivanja privilegija za određena polja i vrste u okviru tabela (to jest, koncept pogleda korisnika kad što je definisano u SQL jeziku). Za posebno velike baze podataka može da bude poželjno da se omogući pojedinim korisnicima da neke od svojih privilegija dodeljuju drugim korisnicima kao na primer davanja prava rukovodiocu da dodeljuje prava pristupa saradnicima koji njemu direktno podnose izveštaje.

Performanse

Kako LAN DBMS sistemi uobičajeno startuju kao način za povezivanje baza podataka pojedinih korisnika bitno je osigurati da se na performanse pojedinih korisnika ne utiče negativno postavljajući ta baza u mrežni red. Performanse pojedinog korisnika uobičajeno su određena time koliko dobro DBMS upravlja U/I operacijama, algoritmima za optimizaciju upita i možda najbitnije od svega stepenom minimizacije saobraćaja na LAN-u (kod većine LAN sistema mreža je uobičajeno kritično usko grlo). Da bi se osiguralo da performanse pojedinačnog korisnika baza podataka budu maksimizirane, poželjno je da aplikativni program bude na istom fizičkom čvoru kao i baza podataka, umesto da se nalazi na posebnom čvoru.

U slučaju kada se baza podataka nalazi na posebnom čvoru bitno je da se minimizira saobraćaj podataka između aplikativnog čvora i čvora baza u cilju maksimiziranja performansi. Ključni faktor koji utiče na performanse LAN DBMS-a je gde se vrši „razdvajanje" između čvora servera i aplikativnog čvora.



Na slici 3 prikazane su tipične komponente DBMS sistema i naznačene su tri različite tačke gde može da se izvrši razdvajanje između softvera servera i aplikativnog softvera. U tabeli 2 je za svaki od tri gornja scenarija (respektivno nazvanih server datoteka, server tabela i server baze podataka) prikazan broj zapisa koji se šalju kroz mrežu za tri različite vrsta upita. Istraživanja su pokazala da je pristup sa serverom baza podataka efikasniji od pristupa sa serverom datoteka u uslovima rada u lokalnoj računarskoj mreži.

Upit	Broj zapisa poslat u LAN-u Server datoteka	Server BP
1 Selekcija 10 zapisa od 10 000	10 000	10
2 Selekcija 10 zapisa iz spoja 1000×1000	1 000 000	10
3 Izračunavanje 10 sumarnih zapisa od 1000	1000	10

Tabela 2 Opterećenje mreže

Da bi efikasno upravljao konkurentnim zahtevima korisnika, LAN DBMS može da podržava dva načina rada. Kod prvog načina vrši se blokiranje pristupa određenim korisnicima dok drugi rade sa bazom podataka smanjujući na taj način konkurentnost rada ali osiguravajući konzistentnost podataka. Drugi način je da se ne vrši zabrane pristupa bazi podataka, maksimizirajući na taj način konkurentnost rada ali i ne garantujući konzistentnost rezultata. Izbor stavki podataka nad kojima će se postavljati zabrana i mogućnost variranja nivoa zabrana, često mogu biti kritični prilikom obezbeđenja optimalnih performansi za širok opseg opterećenja u mreži. Dok serveri datoteka obezbeđuju zabrana pristupa na nivou bajtova upravljanje kompleksnim postavljanjem zabrana na nivou objekata baza podataka (to jest zapisa, stranica, tabela) efikasnije se radi na serveru baze podataka.

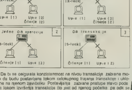
Konzistentnost

Kod svih multikorisničkih sistema ako je promena koju u bazi podataka vrši jedan korisnik vidljiva drugom korisniku u trenutku kada se promena vrši tada su podaci u bazi nekonzistentni. U cilju sprečavanja tog tipa nekonzistentnosti većina DBMS sistema garantuje zabrana nad stavkama podataka u trenutku kada se oni očitavaju ili modifikuju. Izbor koji će se tipovi zabrana podržavati (raspodeljena ili ekskluzivna) kao i trajanja zabrana (tokom jedne akcije nad bazom podataka ili za vreme trajanja transakcija) određuje nivo konzistentnosti raspoloživ u okviru datog DBMS-a. Dobro konstruisani mrežni DBMS treba da ponudi aplikativnim programima sva četiri nivoa konzistentnosti navedena u **tabeli 3.**

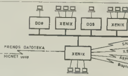
Nivo	Opis
1	**Sprečava da se podatak očitava dok ga neko drugi ažurira**
2.	**Sprečava da se očitavaju sve stavke podataka koje neko ažurira, ali to još uvek nije potvrdio u okviru tekuće transakcije**
3	**Sprečava da se podatak ažurira dok ga neko očitava**
4	**Sprečava da se ažuriraju sve stavke podataka koje neko očitava i može ponovo da očitava u okviru tekuće transakcije**

Tabela 3 Nivoi konzistentnosti

Nivoi konzistentnosti su kumulativni to jest svaki nivo obezbeđuje dodatni nivo zaštite nad onim koje obezbeđuju niži nivoi. Nivoi od 1 od 3 zahtevaju da DBMS implementira koncept transakcije. Pod transakcijom u ovom slučaju se podrazumeva nedeljiva jedinica rada, odnosno skup akcija nad bazom podataka. Kada se potvrdi izvršenja transakcije to podrazumeva da je baza podataka u logički konzistentnom stanju. **Na slici 4** ilustrovani su tipovi zabrana koje se postavljaju za svaki od različitih tipova nivoa konzistentnosti.



Da bi se osigurala konzistentnost na nivou transakcija zabrana moraju da budu postavljena tokom celokupnog trajanja transakcije i uklonjene na njenom završetku. Postavljanja zabrane pristupa stavci podataka tokom izvršenja transakcije (to jest od njenog početka pa dok se ne potvrdi) daje kao rezultat veću konzistentnost ali takođe smanjuje konkuretnost rada.



Sl 5 Lokalna mreža sa računarima TIM 030 i XENIX-NET-om

Obezbeđenje svih nivoa konzistentnosti nije još uvek moguće za sadašnju generaciju servera datoteka. Razlog je što serveri vode računa samo o datotekama (to jest o nizovima bajtova) i nemaju informacija o objektima baze podataka (kao što su zapisi stranice ili transakcije). Ovo je još jedan razlog za korišćenje modela servera baze podataka na personalnim računarima umesto modela servera datoteka

LOKALNE MREŽE RAČUNARA SERIJE TIM I DISTRIBUIRANE BAZE PODATAKA

Računari serije TIM (TIM 022 TIM 030 TIM 600) koriste standardne operativne sisteme MS—DOS UNIX i XENIX. Povezivanje ovih računara u mrežu moguće je korišćenjem sva tri IEEE standarda 802 3 802 4 i 802 5 Ovo je urađeno koristeći elemente za povezivanje najpoznatijih proizvođača a sa Ethernet, Arcnet Starlan i Token-Ring karticama. Moguće je takođe koristiti mrežni softver različitih proizvođača kao i različite servere datoteka odnosno servere baze podataka. Među velikim brojem različitih kombinacija korišćenog hardvera i softvera razmotrićemo sisteme koji koriste SCO XENIX-NET i sisteme koji koriste „Novell"-ov Advanced Netware

Mreže pod XENIX-NET-om

SCO XENIX NET je softverski paket koji omogućava da se u istu lokalnu mrežu uključe računari koji koriste MS—DOS kao i oni koji koriste SCO XENIX. Ovaj paket omogućava raspodelu datoteka koje se nalaze na serverima datoteka u lokalnoj mreži kao i raspodelu štampača i masovnih memorija SCO XENIX—NET distribuirani sistem datoteka (DFS) integriše sve XENIX i DOS servere datoteka u jedinstven sistem. Bilo koja XENIX aplikacija može transparentno da pristupi svakoj datoteci na bilo kom XENIX sistemu u mreži (sl. 5).

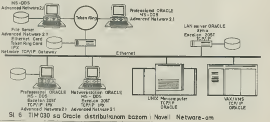
SCO XENIX—NET podržava NETBIOS i kompatibilan je sa DOS mašinama koje koriste IBM PC Network softver ili druge MS—NET derivative. Korisnici na MS—DOS mašinama vide udaljene XENIX resurse kao da su to njihovi diskovi ili štampači. Takođe je obezbeđena sinhronizacija simultanih zahteva za lokovanjem zapisa i datoteka od strane MS—DOS i XENIX-a nad istom datotekom

Na ovaj način omogućeno je projektantima sistema da grade modularna pouzdana sistema. Konfiguracije se jednostavno proširuju u malim inkrementima od jednog XENIX multikorisničkog sistema na veći broj sistema sa više desetina korisnika povezanih u jedinstveni sistem

Mreže pod NOVELL-om

Jedno od najnovijih rešenja koje se koristi na računarima serije TIM za rad sa distribuiranim bazama podataka u mreži je ORACLE DBMS pod NOVELL NetWare-om. ORACLE podržava Excelan-ov TCP/IP protokol kao i NOVELL-ov SPX protokol. Na taj način NOVELL-ovi serveri datoteka i ORACLE serveri baze podataka mogu da budu povezani u istu mrežu kao članovi distribuirane baze podataka

Korišćenjem gejtveja moguće je povezati Ethernet LAN sa bilo kojom drugom mrežom koja podržava NOVELL-ov softver. ORACLE korisnik na TIM-u 030 uključen u token ring LAN može da pristupi ne samo lokalnom NetWare serveru datoteka već takođe bilo kom od ORACLE servera baze podataka (sl. 6)

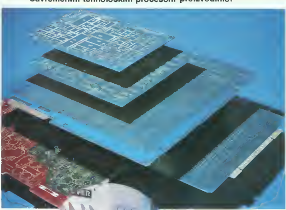


Sl 6 TIM 030 sa Oracle distribuiranom bazom i Novell Netware-om

SA NAMA BRŽE U EVROPU I 21.VEK

Ei ELEKTRONSKA INDUSTRIJA FABRIKA ŠTAMPANIH KOLA

Savremenim tehnološkim procesom proizvodimo:





- **JEDNOSLOJNE I DVOSLOJNE ŠTAMPANE PLOČE**

— VELIKE SERIJE (100, 1 000 i više komada)
Dinamika isporuke prema potrebama kupca

Proizvodnja svih vrsta ploča može da se organizuje na bazi dokumentacije poručioca pripremljene klasičnim postupcima, ili savremenim računarskim sistemima za projektovanje

Proizvedene ploče zadovoljavaju standarde

JUS IEC SNO MIL

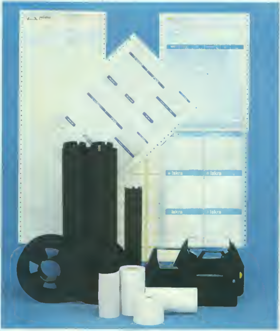
100% električno testirane ploče

- **VIŠESLOJNE ŠTAMPANE PLOČE**
(12, 14 do 24 sloja)
Rok izrade: 15 dana

BESPLATNO NUDIMO PROTOTIPNE KOLIČINE VIŠESLOJNIH ŠTAMPANIH PLOČA OD 16 DO 24 SLOJA. PONUDA VAŽI DO KRAJA GODINE.

- **FLEKSIBILNE ŠTAMPANE PLOČE I PLOČE ZA SMD TEHNIKU**

Informacije: Ei FABRIKA ŠTAMPANIH KOLA


Iskra
Iskra
Iskra
Iskra